

THE
GEORGE
WASHINGTON
UNIVERSITY

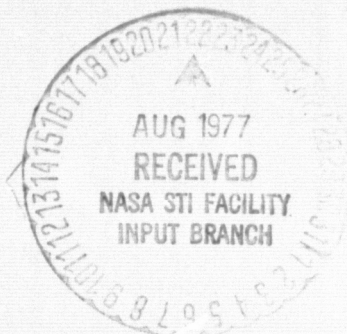
(NASA-CR-153898) PROVING THE CORRECTNESS OF
THE FLIGHT DIRECTOR PROGRAM EADIFD, VOLUME 1
(George Washington Univ.) 127 p HC A07/MF
A01 CSCL 09B

N77-29826

Unclass

G3/61 42984

STUDENTS FACULTY STUDY R
ESEARCH DEVELOPMENT FUT
URE CAREER CREATIVITY CO
MMUNITY LEADERSHIP TECH
NOLOGY FRONTIER DESIGN
ENGINEERING APP ENO
GEORGE WASHINGTON UNIV



SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

Proving the Correctness of the Flight Director
Program EADIFD

Volume One

By

Frank J. Lee

Department of Electrical Engineering

&

Computer Science

School of Engineering & Applied Science

The George Washington University

July 1977

TABLE OF CONTENTS

	<u>Page No.</u>
General Description of EADIFD	1
C4000 Computer Data Representation and Instructions	2
The C4000 Assembly Language Verifier	6
The Verification Condition Generator	8
The Proof Checker	12
The Proof of EADIFD	15
Input and Output Files	20

ACKNOWLEDGMENTS

REFERENCES

APPENDICES:

- I. Listing of EADIFD
- II. Specifications of EADIFD
- III. (1) Listing of the Verifier Program
 - (2) Theorem Library
 - (3) Abbreviation Definitions
 - (4) Proofs
 - (a) Paths 1-29 (Bound separately; see Volume Two)
 - (b) Paths 30-58 (Bound separately; see Volume Three)

PROVING THE CORRECTNESS OF THE FLIGHT DIRECTOR PROGRAM EADIFD

General Description of EADIFD

EADIFD is written in Litton C4000 symbolic assembly language for execution on the C4000 airborne computer, which performs navigation and guidance computations. Developed at the NASA Langley Research Center, EADIFD has 272 assembly instructions and is a subprogram of a navigation and guidance program. A listing of EADIFD is given in Appendix I. The C4000 computer, a general purpose computer with 32K memory, a 24-bit word size, parallel fractional machine, is a part of the navigation/guidance system, which, in turn, is a subsystem of an RSFS (Research Support Flight System) in a modified Boeing 737 airplane.

Basically, the lateral control law and vertical control law are implemented in EADIFD using earth-referenced navigation parameters such as flight path angle, track angle, and ground speed computed by the C4000 computer. The lateral control law is a second order differential equation of lateral displacement from localizer beam, and the vertical control law is a first order differential equation of vertical displacement from localizer beam.

The purpose of EADIFD is to generate pitch and roll command signals to be used by the flight control computers in an RSFS and by the autothrottle servo, for more efficient

and acceptable operations in the terminal area of the airspace system. Two binary state variables are also set to indicate the states of the corresponding commands at various exits.

ORIGINAL PAGE IS
OF POOR QUALITY

C4000 Computer Data Representation and Instructions

The C4000 data word length is 24 bits. Two's complement arithmetic is used in the binary representation of fixed point numbers. In C4000 assembly instructions, a decimal operand contains a fraction and a binary point indicator, which is a "B" followed by a decimal number ranging from 0 through 23. This indicator tells the assembler where the binary point (counting from the left) is to fall within the binary data word. (B0 means that the binary point is just to the right of the sign bit.) For a quantity scaled Bn, if A is the machine number and X is the number in the problem, then $X = A * 2^{**n}$. Each quantity in the problem is scaled down to the range from -1. to 1. with 1. excluded, to be stored in the machine. An octal operand is specified by placing the letter "O" in front of the octal number.

Since the C4000 computer is a fractional binary machine, the location of the binary point during the computations must be kept track of. The location of the binary point in a data word is called the scale of the data word. Where decimal operands are concerned, addition and subtraction must involve quantities with equal scales, and

multiplication and division will produce results with new scales. In order to handle these consistently, the verifier treats all actual quantities with scale Bn in registers and memory locations as integers, and regards all represented quantities as having been multiplied by $2^{(23-n)}$. After this transformation, all the quantities in EADIFD can be manipulated with integer arithmetic. For example, we could have $X*2^{**S}+Y*2^{**S} = (X+Y)*2^{**S}$ and $X*2^{**S1}/Y*2^{**S2} = (X/Y)*2^{**(S1-S2)}$.

The complete list and descriptions of the standard C4000 assembly instructions can be found in [1]. Those instructions used in EADIFD and their algebraic equivalents are listed as follows:

ABS	\$AC=ABS (\$AC)
ADD Y	[R (\$AC+Y)]; \$AC=\$AC+Y
	or \$AC=N (\$AC+Y)
ADS Y	[R (\$AC+Y)]; \$AC=\$AC+Y; Y=\$AC
	or \$AC=N (\$AC+Y); Y=\$AC
CMA	\$AC=-\$AC
DIV Y	[ABS (\$AC).LT.ABS (Y)]; \$AC=\$AC/Y
LDA Y	\$AC=Y
LDQ Y	\$MQ=Y
MPY Y	\$MQ=MULT (\$AC*Y); \$AC=.UNDEFINED.
SAL K	[R (\$AC*2**K)]; \$AC=\$AC*2**K
SAR K	\$AC=\$AC/2**K
SEX Y	Y=\$RA
SLL K	[DIR (\$AQ*2**K)]; \$AQ=\$AQ*2**K
STA Y	Y=\$MQ

```

STQ Y      Y=$MQ
SUB Y      [R($AC-Y)]; $AC=$AC-Y
           or  Y=N($AC-Y)
TNZ L      $RA=*; IF ($AC.NE.0) GO TO L
TRA L      $RA=*; GO TO L
TRM L      $RA=*; IF ($AC.LT.0) GO TO L
TRP L      $RA=*; IF ($AC.GE.0) GO TO L
TRZ L      $RA=*; IF ($AC.EQ.0) GO TO L

```

with:

```

$AC -- Accumulator
$MQ -- Quotient Register
$AQ -- $AC and $MQ, considered as a double register
$RA -- Cell 20 (octal)
* -- "the location of this instruction word" + 1.

```

The function MULT takes the least significant bits of the resulting double length product as its value. The function $N(x) = \text{if } x \geq 2^{23} \text{ then } x - 2^{24} \text{ else if } x < -2^{23} \text{ then } x + 2^{24} \text{ else } x$. The predicate $R(Y)$ is equivalent to $(-2^{23} \leq Y < 2^{23})$, and the predicate $DIR(Z)$ is equivalent to $(-2^{47} \leq Z < 2^{47})$.

Both direct addressing and indirect addressing are used to fetch an operand from memory. Indirect addressing is indicated by appending a "*" to an instruction. For example, if an instruction TRA* L is used at the end of a subroutine, it will cause a transfer to the return address in cell 20 (octal), loaded by the corresponding SEX L instruction at the beginning of the subroutine.

Restrictions, which must hold in order for an

instruction to be executed properly, are enclosed in brackets and placed in front of the corresponding algebraic equivalent. Consider the arithmetic instruction ADD Y, which adds the contents of the effective memory location Y to the contents of the accumulator, the resulting sum replacing the contents of the accumulator. If both numbers have the same sign, but the sign of the result is opposite, then the machine's capacity has been exceeded. The sum, although determinate, is not correct; the machine will not flag the occurrence of this overflow. In order to prove this situation never arises, we put $R(\$AC+Y)$ as the restriction of the instruction ADD Y, which means that $R(\$AC+Y)$ must be true before we can carry out the operation $\$AC=\$AC+Y$.

However, in EADIFD, some additions and subtractions concerned with quantities representing angles, with scale B0, do produce overflow on purpose. This operation works because adding or subtracting 360 degrees produces no change in an angle; the answer is still correct, whether or not overflow occurred. All problem angles in degrees are scaled down to the range from -1. to 1. with 1. excluded, by dividing 180 degrees. 0 degrees, 90 degrees, -90 degrees, and -180 degrees scaled at B0 will have their machine representation as 000000000, 020000000, 060000000, and 040000000, respectively. 270 degrees has the same machine representation as -90 degrees. Adding 90 degrees to 270 degrees produces 360 degrees, which is equivalent to 0 degrees. The machine result of adding 020000000 to 060000000 is 000000000, which is the same as the machine

representation of 0 degrees scaled at B0, even though an overflow occurred. If this is the case in an instruction, then we apply the N function to its operands in the equivalent. For example, $SAC = N(SAC + Y)$ is the other form of equivalent for the instruction ADD Y when the 360 degree wraparound is wanted.

Subroutine SRLMT is called by EADIFD at various places. It computes the function $SRLMT(SIGNAL, LIMIT) = \text{if } SIGNAL > LIMIT \text{ then } LIMIT \text{ else if } SIGNAL < -LIMIT \text{ then } -LIMIT \text{ else } SIGNAL$. The subroutine has been proved correct [2]. A transfer to SRLMT is interpreted as the assignment $SAC = SRLMT(SAC, \$MQ)$, together with setting the Q register and the three scratch registers (LMTSCR, LMTSCR+1, LMTSCR+2) to undefined quantities. The value of $SRLMT(SAC, \$MQ)$ is obtained by substituting SAC for SIGNAL and \$MQ for LIMIT in the above definition.

The C4000 Assembly Language Verifier

Theoretically, the proof of EADIFD is based on the inductive assertion method of Floyd [3]. A C4000 assembly language program verifier written in SNOBOL has been built with the techniques developed in [2]. It consists of two parts: a verification condition generator and a source language independent proof checker.

The verification condition generator takes source programs with assertions as input, and generates a set of verification conditions as output. The modification index

method [4, 5], which appends a digit to the name of each variable in a program, is used in the construction of verification conditions. The unique identifier thereby created distinguishes different values that a variable may have at different points in a control path. Assertions are predicates which describe the behavior of a program at various points; verification conditions are predicates which describe the behavior of program control paths. A control path starts with an initial assertion, continues with executable instructions, and ends with a final assertion. Verification conditions generated by the system have the following form: Hypotheses .IMP. Conclusions, where .IMP. stands for "implies". A control path is correct if the corresponding verification condition is true. A program is correct if and only if it terminates and all the control paths are correct.

The method that we use to prove that a program terminates is to assign termination sequences, which are non-negative integer expressions, to each assertion point and to show that the termination sequence condition of every control path holds. The termination sequence condition of a control path is a predicate in the verification condition of the control path generated by the system, which says that the value of the termination sequence at the final assertion point of the control path is always lexicographically less than the value of the termination sequence at the initial assertion point of the control path.

The proof checker checks the validity of each

deduction statement in the proof procedure of every verification condition. A deduction statement is essentially a call to a known deduction theorem of certain kinds in the theorem library with actual parameters substituted for corresponding formal parameters. For example, the rule of inference (modus ponens) of first order logic from (A .IMP. B) and A to infer B is stated in the theorem library as:

$$\text{IMP1}(A,B) = (A.\text{IMP}.B), A.\text{IMP}.B$$

A deduction statement like $\text{IMP1}(C,D)$ is a call to the library theorem by the name of IMP1 with C and D corresponding respectively to the formal parameters A and B. The theorem library is a collection of known theorems of different classes such as integer arithmetic, logical theorems, rule of substitutions, etc. The proof checker then examines the substituted theorem to see whether the hypotheses of the theorem are among the hypotheses known at the present time. If so, the conclusion of the theorem is deduced as an intermediate result or as a conclusion which may be used as a hypothesis in further deductions. At the end of a proof procedure, the proof checker will declare the truth or the falsity of the verification condition, depending upon whether all the conclusions in that verification condition have been deduced or not.

The Verification Condition Generator

The verification condition generator for C4000 assembly language programs generates verification conditions

in two steps. The first step is to read in the source program, the specifications, and the scales of the program variables; to parse the source program statement-by-statement; and to build up data bases for later uses. The second step is to generate the verification condition for each control path, starting with an initial assertion and ending with a final assertion, by scanning over the source program.

In the first step, if the syntax of a statement is correct and the operands concerned are compatible in types and in scales, then the source statement, its restriction (if any), its algebraic equivalent, its operand, and the type of the statement are recorded in a two-dimensional source statement array. Otherwise, some error messages are printed out, and the process skips to the next source statement. The reason that source statements are also recorded is that we want to be able to trace every statement and every condition in each control path. However, the entry for source statements in the source statement array can be eliminated, if space available is critical. A symbol table, a symbol index table, a label index table, and an assertion table are constructed during the process by examining the label field and the operand field of a source statement. The symbol table is a two-dimensional array which stores attributes, such as the scale, the type, and the current modification index of each source program variable. The symbol index table associates the name of a variable with its index in the symbol table; this structure is used so

that the symbol table entry can be accessed directly later without searching the array. The label index table associates a label with its location (index) in the source statement array. The assertion table is a two-dimensional array, which stores the text of each assertion and the index of its corresponding assertion point in the source statement array; an assertion at an assertion point is attached to the next source statement in a program by the system, as the predicate which must hold before the execution of this source statement.

The algorithm to generate all the verification conditions of a program is stated step by step below. In what follows:

CI denotes the current index to the source statement.

CS denotes the current source statement corresponding to CI.

CAP denotes the number of current assertion point.

ASSERT denotes the assertion table with ASSERT(CAP,1) as the text of the CAP-th assertion and ASSERT(CAP,2) as the index of the CAP-th assertion point to the source statement array.

HYPOTHESES denotes the list of current hypotheses.

CONCLUSIONS denotes the list of current conclusions.

CONDS denotes the list of current conditions.

STACK is an array used by the generator to keep the indices, the hypotheses, the conclusions, and the entry point of each conditional branch point in the process of

verification condition generation.

We suppose that the program to be verified has an initial assertion at the beginning, and a final assertion at each exit point.

1. Initialize CAP and the modification indices of all variables in the symbol table to be zero. Initialize STACK to be empty.

2. Increase CAP by 1.

3. If CAP corresponds to the last assertion point, stop; the algorithm is complete.

4. Set HYPOTHESES to be ASSERT(CAP,1), CI to be ASSERT(CAP,2), and CONCLUSIONS to be null. Go to step 6.

5. If CI is an assertion point, go to step 9.

6. If CS is an unconditional transfer statement to label L, set CI to be the index value of the label L stored in the label index table, and go to step 5. If CS is a conditional transfer statement to label L, append the current modification indices to those variables occurring in the condition of this conditional transfer statement; assign the result to CONDS; set the entry point EP to be the index value of the label L; save EP, HYPOTHESES and CONDS, CONCLUSIONS, and current modification indices on STACK; concatenate the negation of CONDS to HYPOTHESES; and go to step 8.

7. CS must be some kind of algebraic assignment statement at this point. If CS has a restriction, append the current modification indices to those variables occurring in the restriction, and concatenate the result to CONCLUSIONS.

Append the current modification indices to those variables occurring on the right-hand side of the equal sign in the first algebraic equivalent of CS; increase the current modification index of the variable on the left-hand side of the equal sign by 1; append the updated value to this variable; concatenate the result to HYPOTHESES; and repeat the same process for the rest of the algebraic equivalents consecutively, if there are any left.

8. Increase CI by 1, and go to step 5.

9. Append the current modification indices to those variables occurring in the associated assertion, and concatenate the result to CONCLUSIONS. Output a verification condition as: HYPOTHESES .IMP. CONCLUSIONS. If STACK is empty, then go to step 2. Otherwise, pop up STACK; reset the modification index attribute of each variable in the symbol table, HYPOTHESES, and CONCLUSIONS to the corresponding saved values; load CI with saved EP; and go to step 5.

The Proof Checker

Each verification condition is proved separately. A theorem library of known deduction theorems and a set of abbreviation definitions are the data bases of the proof checker. The proof checker accepts units, each composed of a verification condition and a proof procedure, as input. A proof procedure contains a set of deduction statements, which are calls prepared by a human being to the theorems in

the theorem library, in order to deduce all the statements in the conclusion part of a verification condition, with the corresponding hypothesis part as premises. All the expressions in a proof procedure are stored in suffix notation to make it unnecessary to include parentheses in the process of pattern matching. Pattern matching is conducted internally with strings in suffix notation by the proof checker.

A call expressed as `MACEXP(X)` will cause an expansion of abbreviation `X` to its definition, if `X` is defined in the abbreviation definition data base, and will append the expansion to the current premises. A call like `MACEXP(CEXITO)` will expand `CEXITO` to `(GUID3D.NE.0) .AND. (MDA.GE.HRAD) .AND. (TIMER.EQ.BINTME)` as defined and will append "`CEXITO .EQV. (GUID3D.NE.0) .AND. (MDA.GE.HRAD) .AND. (TIMER.EQ.BINTME)`" to the current premises.

A call like `HYPEXP(Y,Z)`, with `Y` as an abbreviation of a list of assertions and `Z` as a single assertion, causes the system to check to see whether `Z` is an assertion in the definition of abbreviation `Y`. If this is true, then `Z` can be used explicitly as a premise in further deductions.

A typical theorem in the theorem library has the following form: `TN(A1, A2, ..., AN) = H1(A1, A2, ..., AN), H2(A1, A2, ..., AN), ..., HM(A1, A2, ..., AN) .IMP. C(A1, A2, ..., AN)`. Here `TN` is the name; `A1, A2, ..., AN` are the formal parameters; `H1(A1, A2, ..., AN), H2(A1, A2, ..., AN), ..., HM(A1, A2, ..., AN)` are the hypotheses; and `C(A1, A2, ..., AN)` is the conclusion of the theorem. A call

to theorem TN is made by $TN(B1, B2, \dots, BN)$ with $B1, B2, \dots$, and BN as the actual parameters corresponding to the formal parameters $A1, A2, \dots$, and AN .

A call to a known theorem will cause a substitution of the actual parameters in the call statement for the corresponding formal parameters in the theorem definition. Before each substitution, the definition, the formal parameters, and the actual parameters are converted to suffix form. The proof checker then checks the hypotheses of the substituted theorem to see whether each hypothesis is in the premises known at the present time. If so, the conclusion of the substituted theorem is deduced and is appended to the current premises. To make this process more clear, let us consider several theorem calls. For example, the call $SUBST1(A0, B0, X, X.GT.0)$ will produce the premise $B0 > 0$, if we know that $A0 = B0$ and that $A0 > 0$; since the theorem named $SUBST1$ is defined as:

$$SUBST1(P, Q, R, A(R)) = P.EQ.Q, A(P).IMP.A(Q)$$

which says that if $A(R)$ is an assertion, $P = Q$, and $A(P)$ is a premise, then $A(Q)$ is also true. Suppose we need the fact that $(A1+B1)+C1 = A1+(B1+C1)$; we can call the theorem $A2$ with $A1, B1$, and $C1$ as the actual parameters. $A2$ is defined as:

$$A2(P, Q, R) = (P+Q)+R.EQ.P+(Q+R)$$

This is a tautology which states the associative law of addition.

Some simplifications are done automatically by the proof checker. All integer arithmetic assertions, such as

$1+1 = 2$ and $5+2 > 3$, are verified directly without applying any theorems. Equality is considered to be reflexive. If $B2 = A2$ is a hypothesis of a theorem in its definition, and $A2 = B2$ is true, then, in the deduction process of this theorem call, $B2 = A2$ is automatically derived from the fact that $A2 = B2$. Since all pattern matching is conducted internally in suffix form, expressions are free of parentheses and are in a kind of basic canonical form. For example, $((A3+B3)+C3)$ has the same form as $A3+B3+C3$ internally; but $((A3+B3)+C3)$ is not the same as $A3+(B3+C3)$ internally.

At the end of a proof, all the statements in the conclusion part of a verification condition should be deduced at least once in order to complete the proof of correctness for this verification condition.

The Proof of EADIFD

With the specifications (see Appendix II) provided by NASA, EADIFD has been proved correct. Since EADIFD has no loops, the termination of the program is guaranteed. An instruction constancy assertion (ICA) for the problem of program self-modification is in every assertion; this says that the program has not been modified. In the case of EADIFD, ICA remains true throughout the execution of the program, because the program is absolutely pure - i.e., it contains no instructions that can modify it under any conditions. Overflow analysis is done by placing

restrictions on some instructions. The consistency of fixed point scaling is checked by the verification condition generator, which examines the scale of the operands for each kind of arithmetic instruction.

Due to the fact that EADIFD has four exits, four proofs, each with one exit declared as true and others as false, have been completed. Each proof has initial assertion and final assertions as given below.

The first proof has assertions declared as:

Initial assertion: SPEC, CEXIT0, .NOT. CEXIT1,

.NOT. CEXIT2, .NOT. CEXIT3

Final assertion at exit 0: PCMD = PCMD02, RCMD = RCMD01,

RCMVLD = RCMVLD04

Final assertion at exit 1: .FALSE.

Final assertion at exit 2: .FALSE.

Final assertion at exit 3: .FALSE.

The second proof has assertions declared as:

Initial assertion: SPEC, CEXIT1, .NOT. CEXIT0,

.NOT. CEXIT2, .NOT. CEXIT3

Final assertion at exit 0: .FALSE.

Final assertion at exit 1: RCMVLD = 0, PCMVLD = 0

Final assertion at exit 2: .FALSE.

Final assertion at exit 3: .FALSE.

The third proof has assertions declared as:

Initial assertion: SPEC, CEXIT2, .NOT. CEXIT0,

.NOT. CEXIT1, .NOT. CEXIT3

Final assertion at exit 0: .FALSE.

Final assertion at exit 1: .FALSE.

Final assertion at exit 2: PCMD = PCMD03, PCMVLD = 0,

RCMVLD = RCMVLD04

Final assertion at exit 3: .FALSE.

The fourth proof has assertions declared as:

Initial assertion: SPEC, CEXIT3, .NOT. CEXIT0,

.NOT. CEXIT1, .NOT. CEXIT2

Final assertion at exit 0: .FALSE.

Final assertion at exit 1: .FALSE.

Final assertion at exit 2: .FALSE.

Final assertion at exit 3: PCMD = PCMD03, RCMD = RCMD01,

PCMVLD = PCMVLD02, RCMVLD = RCMVLD04

The definitions of CEXIT0, CEXIT1, CEXIT2, and CEXIT3, together with the basic conditions leading up to them are listed as follows:

CEXIT0 = CVCRUZ .AND. TEXTIT0

CVCRUZ = CFD112 .AND. TVCRUZ

TEXTIT0 = (GUID3D.NE.0) .AND. (MDA.GE.HRAD) .AND. (TIMER
.EQ. BINTME)

CFD112 = .NOT. CEXIT1

TVCRUZ = (LOCFD06.EQ.0) .OR. (GSFD02.EQ.0) .AND.
(GSBSNS.EQ.0)

CEXIT1 = (TMEFLG.EQ.0) .OR. CCRUZFD .AND. (GUID2D.EQ.0)

CCRUZFD = CFDIN1 .OR. CFD2 .AND. (LOCFD03.EQ.0)

CFDIN1 = (TMEFLG.NE.0) .AND. T1CRUZFD

CFD2 = CFD4 .AND. (LBS.EQ.0) .OR. CFDIN2 .AND. .NOT.
TFD3 .AND. (GUID2D.NE.0)

T1CRUZFD = (AND(PCUMWD, T12).EQ.0) .OR. (ILSON.EQ.0)
.OR. (GRUND.NE.0)

```

CFD4 = CFDIN2 .AND. (TFD3 .OR. (GUID2D.EQ.0))
CFDIN2 = (TMEFLG.NE.0) .AND. .NOT. T1CRUZFD
TFD3 = (ABS(YETA01)-LIMLOC.LT.0) .OR. N(ABS(PSIE01)-
      DEG2) .LT.0)
CEXIT2 = CFD15 .AND. (GSVLD.EQ.0) .OR. CVCRUZ .AND.
      TEXTIT2
CFD15 = CFD11 .AND. .NOT. TVCRUZ
TEXTIT2 = (GUID3D.EQ.0) .OR. TVCRUZ3 .AND. (PCMVLD.NE.0)
CFD52 = CLANDEFD .AND. .NOT. TFD7
CFD8 = CFD7 .AND. (HRAD.LT.A1200)
CFD9 = CCRUZFD .AND. (GUID2D.NE.0)
CFD10 = CFD9 .OR. CFD7 .AND. (HRAD.GE.A1200)
CLANDEFD = .NOT. CFDIN1 .AND. CFD2 .AND. (LOCFD03.NE.0)
      .OR. CFD42
TFD7 = (YETA01*PSIE01.GT.0) .OR. (LOCSTRO1*PSIE01.LT.0)
      .OR. (N(ABS(PSIE01)-DEG20).LT.0)
CFD7 = CLANDEFD .AND. TFD7
CFD42 = CFD4 .AND. (LBS.NE.0) .AND. (LOCFD03.NE.0)
CEXIT3 = CFD15 .AND. (GSVLD.NE.0) .OR. CVCRUZ .AND. TEXTIT3
TEXTIT3 = (GUID3D.NE.0) .AND. ((MDA.LT.HRAD) .OR. (TIMER
      .NE. BINTME) .AND. (PCMVLD.EQ.0))

```

(Here the symbol "=" stands for "is equivalent to".)

The definitions of RCMD01, RCMVLD04, PCMD01, PCMD02, PCMD03, PCMVLD02 are:

```

RCMD01 = SRLMT(N(RLFD04-ROLL), DEG25)
RLFD04 = N(SRLMT(n(AC02-RLFD02)/2**2/TAU3, DEG4)*DELTA+
      RLFD02)
AC02 = .IF. CFD62 .THEN. A0 .ELSE. .IF. CFD8 .THEN.

```

```

        SRLMT(LOCSTR01, DEG15) .ELSE. SRLMT(AC01, DEG25)
AC01 = .IF. CFD9 .THEN. LATSTR .ELSE. LOCSTR01
LOCSTR01 = -(SRLMT(N(TP205-TP103), LIM180)*2**3)
TP103 = .IF. (LOCOC04.EQ.0) .THEN. TP103A .ELSE. TP103B
TP103A = N(N(YETA01/PERIOD/PERIOD*K5-TP102)/TAU1*DELTAT/
        2**3+TP102)
TP103B = N(N(SRLMT(YETA01, LIMY)/PERIOD/PERIOD*K5-TP102)
        /TAU1*DELTAT/2**3+TP102)
TP205 = .IF. CFD52 .THEN. N(TP203-TP702) .ELSE. TP203
LOCOC04 = .IF. CFD3 .THEN. SET .ELSE. LOCOC02
YETA01 = MULT(LOCDEV, K1)
TP102 = .IF. CFDIN3 .THEN. 0 .ELSE. TP1
TP203 = GS/PERIOD*K6*PSIE01*KDAMP*2**2
TP702 = .IF. CFDIN3 .THEN. 0 .ELSE. TP7
LOCOC02 = .IF. CFDIN3 .THEN. 0 .ELSE. LOCOC
PSIE01 = N(TK-RWYHDG)
RCMVLD04 = .IF. CFD12 .THEN. SET .ELSE. 0
PCMD01 = SRLMT(N(-(TP603*FDBKG*2**3)+GAMCMD01), DEG6)
TP603 = N(PITCH-TP502)
GAMCMD01 = N(LGAMC03-GAMMA)
TP502 = .IF. CFDIN3 .THEN. 0 .ELSE. TP5
LGAMC03 = .IF. TFD155 .THEN. LGAMC04 .ELSE. TLGAMC0
LGAMC04 = .IF. TFD156 .THEN. TLGAMC1 .ELSE. TLGAMC2
TLGAMC0 = N(SRLMT(N(A0-LGAMC02)/TAU5/2**3, LIMPR)*DELTAT
        +LGAMC02)
TLGAMC2 = N(SRLMT(N(GAMMAC01-LGAMC02)/TAU5/2**3, LIMPR)*
        DELTAT+LGAMC02)
TLGAMC1 = N(SRLMT(N(-N(N(N(GAMMAC01+GSA)+GSA)-GAMMAC01)

```

-LGAMC02)/TAU5/2**3,LIMPR)*DELTAT+LGAMC02)

```
LGAMC02 = .IF. CVLAND1 .THEN. GAMMA .ELSE. LGAMC
PCMD02 = N(N(SRLMT(SRLMT(VERSTR, C5B15)/VGS/2**2/KHD/PIB2,
DEG6)-PCMD)/C3B2*DELTAT/2**2+PCMD)
PCMD03 = .IF. CVCRUZ2 .THEN. PCMD02 .ELSE. PCMD01
PCMVLD02 = RCMVLD04
```

Abbreviations such as SPEC, CEXIT0, CFD12, and LOCFD06 are used to make assertions more readable and to save space. Appendix III contains all the proofs, and the complete set of abbreviations, as well as their definitions.

Input and Output Files

The verification condition generator and the proof checker are used separately; the verification conditions of a program have generated first and then to be proved.

In order to use the verification condition generator, the programmer has to provide the input-output specifications, the scales of program variables, the abbreviation definitions, and a set of assertions and termination sequences at specific points in the program which must have at least one assertion and one termination sequence for each loop. An assertion has the same form as a Boolean expression written in Fortran, with .EQ., .NE., .GT., .LE., .LT., .GE. as the relational operators and .NOT., .AND., .OR., .IMP., .EQV. as the logical operators. .TRUE. and .FALSE. are the logical constants. For example, TMEFLG.GE.0 and ILSON.NE.0.AND.3RUND.NE.0 are the typical

assertions used in the proof of EADIFD. A termination sequence is a sequence of non-negative integer expressions associated with certain assertion points. Functions such as $ABS(P)$, $MAX(P,Q)$, and $AND(P,Q)$ are used in the usual sense. A user-defined function must be declared in the theorem library before using it. The system also handles array references and has special assertions to declare it; although no array reference occurred in EADIFD. The output contains a listing of the program with assertions, the specifications, the scalings, and a set of verification conditions. An option can be chosen to punch the verification conditions into cards for the preparation of the input to the proof checker.

A proof procedure, a set-up to prove a verification condition, is a basic input unit to the proof checker. The proof checker takes the file of the theorem library, the specifications, the abbreviation definitions, and a number of proof procedures as input. A proof procedure consists of the hypotheses and the conclusions of a verification condition, the names of the theorems to be used, and a set of theorem calls. The output is in the following format:

COMMENTS

***(if any)

***
.
.
.

HYPOTHESES

H1:

H2:

.
.
.

Hm:

CONCLUSIONS

C1:

C2:

.
.
.

Cn:

THEOREMS USED

T1: (with theorem definition expanded)

T2:

.
.
.

Tk:

PROOF

THEOREMS

HYPOTHESES

CONCLUSIONS

(theorem call)	(the hypothesis part of the theorem call)	(the conclusion part of the theorem call)
----------------	---	---

.	.	.
.	.	.
.	.	.

***** Q. E. D. ***** (or the error message which
states the reason why this proof is not completed)

A distinct identification label which has the form of "Hn" or "Cn" with n as an integer is assigned by the system for each hypothesis or conclusion, respectively. When an intermediate result is deduced from a theorem call, the result is identified by a label of the form "Ln". If a conclusion is deduced, the conclusion part in the output of this theorem call is denoted by a label of the form "*Cn", where the "*" is used to indicate that the nth conclusion of this verification condition has been proved. The use of these labels is for the purpose of making each proof step clearer in the output of the proof checker.

ACKNOWLEDGMENTS

The research herein reported was partially supported by National Aeronautics and Space Administration Grant NSG-1170 and by National Science Foundation Grant DCR73-03431-A01.

REFERENCES

1. Litton Aero Products, C-4000 programming manual, Document No. 70-081.10, Litton Aero Products, Woodland Hills, Calif., September 1972.
2. Maurer, W. D., An IBM assembly language program verifier, Proc. 16th Annual Technical Symposium, ACM Washington Chapter, National Bureau of Standards, Gaithersburg, Md., June 1977, pp. 139-146.
3. Floyd, R. W., Assigning meanings to programs, Proc. Symp. Applied Math. 19 (Mathematical Aspects of Computer Science), American Mathematical Society, Providence, R. I., 1967, pp. 19-32.
4. Maurer, W. D., The modification index method of generating verification conditions, Proc. 15th Annual Southeast Regional ACM Conference, Biloxi, Miss., April 1977, pp. 429-440.
5. Maurer, W. D., A new method of generating verification conditions, Proc. 1977 Conf. on Information Sciences and Systems, Johns Hopkins Univ., Baltimore, Md., March 1977, pp. 128-133.

APPENDIX I

C-4000 ASSEMBLY				15 JULY 1975 1 RSIS C4003 F110 750719				PAGE 210	
FLAGS	LOC	MEM-WORD	OP	M	LABEL	OPC	OPERAND	COMMENTS	SEQUENCE LINE
					****			PUTIN	7807
					*****			*****	7808
					FLIGHT DIRECTOR ROUTINE				7809
					*****				7810
									7811
									7812
55427	35057301	35	57301	EADIFD	SEX	EFDKTN			7813
55430	01000733	1	233	LDA	TIMEFLG			REAL TIME S	7814
55431	10055623	10	55623	TRZ	EXIT1				7815
55432	01040165	1	40165	LDA	PCUMWD			EXTRACT BIT FOR	7816
55433	16003636	16	3636	ANA	T12			ILS SELECTED ON	7817
55434	12057317	12	57317	STA	MPILS			EADI MCDE PANEL	7818
55435	01057317	1	57317	LDA	MPILS			***** USE LAND MODE WHEN AVAILABLE *****	7819
55436	10055443	10	55443	TRZ	**5				7820
55437	01000070	1	70	LDA	ILSON			ILS TUNED S	7821
55440	10055443	10	55443	TRZ	**3				7822
55441	01057305	1	57305	LDA	GRUND			NOT GO AROUND S	7823
55442	10057347	10	55447	TRZ	**5				7824
55443	14057323	14	57323	STZ	STAPT				7825
55444	14057313	14	57313	STZ	LUCFD				7826
55445	14057307	14	57307	STZ	GSFD				7827
55446	05055621	5	55621	TRA	CRUZFD				7828
55447	057323	1	57323	LDA	START			FIRST PASS S	7829
55450	11055665	11	55465	TRZ	FD1				7830
55451	01003477	1	3477	LDA	SET			START MPILS X ILSON X (-GRUND)	7831
55452	12057323	12	57323	STA	START				7832
55453	14057325	14	57325	STZ	TP1			INITIALIZE FOR	7833
55454	14057326	14	57326	STZ	TP2			FOR FIRST PASS	7834
55455	14057327	14	57327	STZ	TP3				7835
55456	14057330	14	57330	STZ	TP4				7836
55457	14057331	14	57331	STZ	TP5				7837
55460	14057332	14	57332	STZ	TP6				7838
55461	14057333	14	57333	STZ	TP7				7839
55462	14057313	14	57313	STZ	LUCFD				7840
55463	14057314	14	57314	STZ	LUCOC				7841
55464	14057302	14	57302	STZ	FLARE				7842
55465	01003373	1	73	LDA	GPLCCD				7843
55466	27056053	27	56053	MPY	K1				7844
55467	12057334	12	57334	STA	YETA				7845
55470	01000341	1	341	LDA	TK				7846
55471	04020021	4	20021	SUB	RWYHDG			COMPUTE PSI	7847
55472	12057321	12	57321	STA	PSIE			ERROR	7848
55473	37000400	37	0	ABS					

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX I

L-4000 ASSEMBLY

15 JULY 1975 : RSFS C4KB3 FT10 750715

PAGE 211

FLAGS	LOC	MEM-WORD	I OP	M	LABEL	OPC	OPFRAND	COMMENTS	SEQUENCE	LINE
	55474	04005337	4	5307		SUB	DEG2	WITHIN LIMITS \$		7849
	55475	06055507	6	55507		TRM	FD3			7850
	55476	01057334	1	57334		LDA	YETA			7851
	55477	37000600	37	0		ABS				7852
	55500	04056066	4	56066		SUB	LIMLOC	LAT DEV WITHIN LIMITS \$		7853
	55501	06055507	6	55507		TRM	FD3			7854
	55502	01001021	1	1021		LDA	GUID2D	2D GUIDANCE ON \$		7855
	55503	10055511	10	55511		TRZ	FD4			7856
	55504	01057313	1	57313	FD2	LDA	LOCED	BRANCH LOC/CRUISE F.D.		7857
	55505	11055521	11	55521		TRZ	LANDED			7858
	55506	05055621	5	55621		TRA	CRUZF0			7859
	55507	01003477	1	3477	FD3	LDA	SET			7860
	55510	12057314	12	57314		STA	LUCOC			7861
	55511	01000455	1	455	FD4	LDA	LBS	LOC BEAM SENSED \$		7862
	55512	10055504	10	55504		TRZ	FD2			7863
	55513	01057313	1	57313		LDA	LOCED	TEST FOR FIRST PASS		7864
	55514	11055521	11	55521		TRZ	LANDED			7865
	55515	01003477	1	3477		LDA	SET	INITIALIZE		7866
	55516	12057313	12	57313		STA	LOCED	RLFD		7867
	55517	01000455	1	455		LDA	ROLL	AS FD BARS		7868
	55520	12057322	12	57322		STA	RLFD	COME UP		7869
	55521	01057314	1	57314	LANDED	LDA	LUCOC	LOCALIZER ON COURSE		7870
	55522	10055527	10	55527		TRZ	FD5	NO, SKIP COURSE CUT LIMITER		7871
	55523	01057334	1	57334		LDA	YETA			7872
	55524	02056070	2	56070		LDQ	LIMY			7873
	55525	05042555	5	4255		TRA	SPLMT			7874
	55526	05055530	5	55530		TRA	**2			7875
	55527	01057334	1	57334	FD5	LDA	YETA	YETA OR LIMITED YETA		7876
	55530	34056072	34	56072		DIV	PERIOD			7877
	55531	34056072	34	56072		DIV	PERIOD			7878
	55532	27056057	27	56057		MPY	K5	COMPUTATION OF		7879
	55533	04057325	4	57325		SUB	TP1			7880
	55534	34056073	34	56073		DIV	TAU1	TP1 (YETA X K5 / PER**2) / ((TAU1 X SC-1)		7881
	55535	27000236	27	236		MPY	DELTAT			7882
	55536	37000203	37	3		SAR	3			7883
	55537	26057325	26	57325		ADS	TP1			7884
	55540	01000336	1	336		LDA	GS	*****		7885
	55541	34056072	34	56072		DIV	PERIOD			7886
	55542	27056060	27	56060		MPY	K6	COMPUTATION OF		7887
	55543	27057321	27	57321		MPY	PSIE			7888
	55544	27056052	27	56052		MPY	KDAMP	(GS X PSIE X KDAMP X K6)/PERIOD		7889
	55545	37000102	37	2		SUL	2			7890

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX I

ORIGINAL PAGE IS
OF POOR QUALITY

C-4000 ASSEMBLY				15 JULY 1975 : RSFS C4KB3 FT10 750715				PAGE 212		
FLAGS	LOC	MEM-WORD	I DP	M	LABEL	CPC	OPERAND	COMMENTS	SEQUENCE	LINE
	55546	12057326	12	57326		STA	TP2			7891
	55547	01100350	1	350		LDA	HRAD	*****		7892
	55550	04056064	4	56064		SUB	K1000			7893
	55551	07055561	7	55561		TRP	FD6			7894
	55552	01057326	1	57326		LDA	TP2			7895
	55553	04057333	4	57333		SUB	TP1	WASHOUT IF		7896
	55554	12057326	12	57326		STA	TP2	HRAD LESS		7897
	55555	27000236	27	236		MPY	DELTAT	THAN 1000 FT		7898
	55556	34056074	34	56074		DIV	TAU2			7899
	55557	37000206	37	6		SAH	6			7900
	55560	26057333	20	57333		ADS	TP1			7901
	55561	01057326	1	57326	FD6	LDA	TP2	*****		7902
	55562	04057325	4	57325		SUB	TP1			7903
	55563	02056365	2	56365		LDQ	LIM180			7904
	55564	05004255	5	4255		TRA	SRLMT			7905
	55565	37000003	37	3		SAL	3			7906
	55566	37000500	37	0		CMA				7907
	55567	12057315	12	57315		STA	LOCSTR			7908
	55570	01057334	1	57334		LDA	YETA	YETA X PSIE		7909
	55571	27057321	27	57321		MPY	PSIE	LE ZERO S		7910
	55572	10055574	10	55574		THZ	**2			7911
	55573	07055606	7	55606		TRP	FD7			7912
	55574	01057315	1	57315		LDA	LOCSTR	LOCSTR X PSIE		7913
	55575	27057321	27	57321		MPY	PSIE	GE ZERO S		7914
	55576	06055606	6	55606		TRM	FD7			7915
	55577	01057321	1	57321		LDA	PSIE	PSIE GT 20		7916
	55600	37000400	37	0		ABS				7917
	55601	04005333	4	5303		SUB	DEG20			7918
	55602	07055606	6	55606		TRM	FD7			7919
	55603	14057330	14	57330		STZ	CAPT			7920
	55604	01002500	1	3500		LDA	AO			7921
	55605	05055631	5	55631		TRA	FD11			7922
	55606	01003477	1	3477	FD7	LDA	SET			7923
	55607	12057330	12	57330		STA	CAPT			7924
	55610	01060350	1	350		LDA	HRAD	HRAD GT 1200s		7925
	55611	04056047	4	56047		SUB	K1200			7926
	55612	06055615	6	55615		TRM	FD8	LFSS THAN 1200		7927
	55613	01057315	1	57315		LDA	LOCSTR			7928
	55614	05055627	5	55627		TRA	FD10			7929
	55615	01057315	1	57315	FD8	LDA	LOCSTR			7930
	55616	02040161	2	40161		LDQ	DEG15			7931
	55617	05004255	5	4255		TRA	SRLMT	LIMIT LOCSTR AT LOW ALT.		7932

APPENDIX I

C-4000 ASSEMBLY				15 JULY 1975 1 RSFS C4K03 FT10 730719				PAGE 213		
FLAGS	LOC	MTT-WORD	I (P)	M	LABEL	OPC	OPERAND	COMMENTS	SEQUENCE	LINE
	55620	05055631	5	55631	TRA	FD11				7933
	55621	01001021	1	1021	CRUZFD	LDA	GUI02D	CRUISE F.O.		7934
	55622	11055626	11	55626	TNZ	FD9		GUI02D ON S		7935
	55623	14000366	14	36	EXIT1	STZ	RCMVLD	BOTH ROLL AND		7936
	55624	14300067	14	67		STZ	PCMVLD	PITCH COMMANDS INVALID		7937
	55625	45057301	5	57301	TRA*	FFDRTN				7938
	55626	01003356	1	3356	FD9	LDA	LATSTR			7939
	55627	02003377	2	3377	FD10	LDQ	DEG25			7940
	55630	05004255	5	4255	TRA	SRLMT				7941
	55631	05057322	4	57322	FD11	SUB	RIFD			7942
	55632	37000202	37	2	SAR	2				7943
	55633	34355015	34	56015	DIV	TAU3				7944
	55634	02005106	2	5306	LDQ	DEG4		LIMIT ROLL RATE		7945
	55635	05004255	5	4255	TRA	SRLMT				7946
	55636	27000236	27	236	MPY	DELTAT				7947
	55637	20057322	20	57322	ADS	RIFD				7948
	55640	04000345	4	345	SUB	ROLL				7949
	55641	02003377	2	3377	LDQ	DEG25				7950
	55642	05004255	5	4255	TRA	SRLMT				7951
	55643	12001057	12	1057	STA	RCMD				7952
	55644	01057313	1	57313	LDA	LCCFD		*****		7953
	55645	10355655	10	55655	TRZ	FD13		ROLL COMMAND		7954
	55646	01300071	1	71	LDA	LCCVLD				7955
	55647	10055657	10	55657	TRZ	FD14				7956
	55648	01057317	1	57317	LDA	MPILS		VALIDITY TEST		7957
	55649	10355657	10	55657	TRZ	FD14				7958
	55652	01301577	1	3477	FD12	SET		SET ROLL		7959
	55653	12300366	12	36	STA	RCMVLD		COMMAND VALID		7960
	55654	05055660	5	55660	TRA	VERTFD				7961
	55655	01001021	1	1021	FD13	LDA	GUI02D			7962
	55656	11055652	11	55652	TNZ	FD12				7963
	55657	14001066	14	66	FD14	STZ	RCMVLD	*****		7964
	55658	01057313	1	57313	VERTFD	LDA	LCCFD	TEST FOR ILS FLT DIR		7965
	55659	10056013	10	56013	TRZ	VERU2		GO TO CRUISE MODE		7966
	55662	01057307	1	57307	VLAND	LDA	GSFD	TEST FOR FIRST PASS		7967
	55663	11055672	11	55672	TNZ	FD15		NOT FIRST PASS		7968
	55664	01057306	1	57306	LDA	GSBSNS		TEST FOR GLIDE SLOPE BEAM SENSED		7969
	55665	10056013	10	56013	TRZ	VERU2		GLIDE SLOPE NOT SENSED	GO TO CRUISE	7970
	55666	01003477	1	3477	LDA	SET		FIRST PASS IN GLIDE SLOPE		7971
	55667	12057307	12	57307	STA	GSFD		SET GSFD HI AND		7972
	55670	01001152	1	1152	LDA	GAMMA		INITIALIZE LGAMP TO ROLL		7973
	55671	12057312	12	57312	STA	LGAMP		WHEN FD BARS COME UP		7974

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX I

(4000 ASSEMBLY

15 JULY 1975 1 NSCS C4KRI 1110 150714

PAGE 214

FLAGS	LOC	MEM-WORD	OP	M	LABEL	OPC	OPERAND	COMMENTS	SEQUENCE	LINE
	55672	0 300074	1	74	FD15	LDA	GPQSDV	*****		
	55673	27056053	27	56053		MPY	K1			7975
	55674	37003110	37	10		SLL	8			7976
	55675	04057327	4	57327		SUB	TP3	COMPUTE GAMMA CMD		7977
	55676	34056076	34	56076		DIV	TAU4			7978
	55677	27000236	27	236		MPY	DELTAT	TP4 1GSDEVXK1/1TAU4S+11XK3/1801+GSA-K4		7979
	55700	37033204	37	4		SAR	4			7980
	55701	20057327	20	57327		AJS	TP3			7981
	55702	34037700	34	3700		DIV	C180BF			7982
	55703	27056055	27	56055		MPY	K3			7983
	55704	03070033	3	20033		AND	GSA	GLICE SLOPE ANGLE		7984
	55705	04056056	4	56056		SUB	K4			7985
	55706	27000500	37	0		CMA				7986
	55707	12057330	12	57330		STA	TP4			7987
	55710	01057302	1	57302		LDA	FLARE	*****		7988
	55711	11055724	11	55724		THZ	FLRALT	FLARE LATCHED S		7989
	55712	01303357	1	357		LDA	HOCF	*****		7990
	55713	03052054	3	56054		ADD	K2	LIMIT HOCF		7991
	55714	06055717	6	55717		TRM	**3			7992
	55715	01056063	1	56063		LDA	K10	TO LESS		7993
	55716	05055720	5	55720		TRA	**2	THAN -3		7994
	55717	01050357	1	357		LJA	HOCF			7995
	55718	0 356061	3	56061		ADD	K7	COMPUTE		7996
	55721	2 56101	27	56101		MPY	TF	HREF		7997
	55722	34056062	34	56062		DIV	K8	IF FLARE		7998
	55723	12057310	12	57310		STA	HREF	NOT LATCHED		8000
	55724	01000350	1	350	FLRALT	LDA	HREF	COMPARE HREF AND HREF		8001
	55725	34057310	34	57310		DIV	HREF	FOR FLARE LATCHING		8002
	55726	12057311	12	57311		STA	KFLR	AND FLARE GAIN COMPUTATIONS		8003
	55727	04003666	4	3666		SUB	MSKL15	HREF/HREF		8004
	55730	07055734	7	55734		TRP	**4	LESS THAN ONE		8005
	55731	01003477	1	3477		LDA	SET	YES, SET		8006
	55732	12057302	12	57302		STA	FLARE	FLARE LATCH		8007
	55733	05055735	5	55735		TRA	**2			8008
	55734	14057302	14	57302		STZ	FLARE	NO, ZERO FLARE LATCH		8009
	55735	01057311	1	57311		LDA	KFLR			8010
	55736	02003666	2	3666		LDQ	MSKL15			8011
	55737	05004255	5	4255		TRA	SRLMT			8012
	55740	37003010	37	10		SAL	8			8013
	55741	27057330	27	57330		MPY	TP4			8014
	55742	04056056	4	56056		SUB	K4			8015
	55743	06055746	6	55746		TRM	**3	*****		8016

APPENDIX I

C-4000 ASSEMBLY				15 JULY 1975 : RSFS C4KH3 FT10 150715				PAGE 215		
FLAGS	LOC	MEM-WORD	I OP	M	LABEL	OPC	OPERAND	COMMENTS	SEQUENCE	LINE
	55744	01003500	1	3500		LDA	A0			
	55745	05057506	5	55756		TRA	**0	LIMIT GAMMA COMMAND		8017
	55746	12057304	12	57304		STA	GAMMAC	BETWEEN ZERO AND MINUS 2*GSA		8018
	55747	03020033	3	20033		ADD	GSA			8019
	55750	03020033	3	20033		ADD	GSA			8020
	55751	06055754	6	55754		TRM	**3			8021
	55752	01057304	1	57304		LDA	GAMMAC			8022
	55753	05055756	5	55756		TRA	**3			8023
	55754	04057304	4	57304		SHB	GAMMAC			8024
	55755	37000500	37	0		CMA		*****		8025
	55756	04057312	4	57312		SUB	LGAMC			8026
	55757	34056077	34	56077		DIV	TAUS	INTEGRATE GAMMA COMMAND		8027
	55758	37000503	37	3		SAK	3			8028
	55759	02056067	2	56067		LQ	LIMPR	LIMIT PITCH RATE		8029
	55762	05004255	5	4255		TRA	SKLMT			8030
	55763	27000236	27	236		MPY	DELTAT			8031
	55764	20057312	20	57312		ADS	LGAMC			8032
	55765	04001152	4	1152		SUB	GAMMA			8033
	55766	12057303	12	57303		STA	GAMCMD			8034
	55767	01000146	1	346		LDA	PITCH	*****		8035
	55770	04057331	4	57331		SUB	TP5	COMPUTE		8036
	55771	12057332	12	57332		STA	TP6	WASHOUT		8037
	55772	27000236	27	236		MPY	DELTAT			8038
	55773	34056130	34	56100		DIV	TAU6	TIME		8039
	55774	37000203	37	3		SAK	3	CONSTANT		8040
	55775	20057331	20	57331		ADS	TP5			8041
	55776	01057332	1	57332		LDA	TP6			8042
	55777	27057351	27	56051		MPY	FDBKG			8043
	56000	37000103	37	3		SLI	3			8044
	56001	37000500	37	0		CMA		*****		8045
	56002	03057303	3	57303		ADD	GAMCMD	COMPUTE PITCH COMMAND		8046
	56003	02050500	2	56050		LQ	DEG6	LIMIT PITCH CMD TO 6 DEG		8047
	56004	05004255	5	4255		TRA	SRLMT			8048
	56005	12001060	12	1060		STA	PCMD			8049
	56006	01000102	1	72		LDA	GSVLD	GS TUNED S		8050
	56007	10055045	10	56045		TRZ	EXIT2			8051
	56010	01000106	1	66	EXIT3	LDA	RCVLD	PITCH COMMAND IS VALID IF		8052
	56011	12000067	12	67		STA	PCVLD	ROLL COMMAND IS VALID		8053
	56012	45057301	5	57301		TRA	EFDPTN			8054
	56013	01001022	1	1022	VCRUZ	LDA	GUID3D	3D PATH IN S		8055
	56014	10056045	10	56045		TRZ	EXIT2	NO, BLANK VERT COMMAND		8056
	56015	01000357	1	3357		LDA	VERSTR	FETCH 3D GUIDANCE STEERING SIGNAL		8057
										8058

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX I

C-4000 ASSEMBLY

15 JULY 1975 1 MSFS C4K03 FT10 750715

PAGE 216

FLAGS	LOC	MEM WORD	OP	M	LABEL	OPC	OPERAND	COMMENTS	SEQUENCE	LINE
	56016	02003710	2	3710	LDQ	C5H15		5 EPS*02 LIMIT		8059
	56017	05004255	5	4255	TRA	SRLMT				8060
	56020	34000317	34	337	DIV	VGS				8061
	56021	37000202	37	2	SAR	2				8062
	56022	34002301	34	3301	DIV	KHD				8063
	56023	34003406	34	3406	DIV	PIB2				8064
	56024	02004050	2	56050	LDQ	DEG6		LIMIT PCMD TO 6 DEG		8065
	56025	05004255	5	4255	TRA	SRLMT				8066
	56026	04001060	4	1060	SUR	PCMD				8067
	56027	34003435	34	3435	DIV	C342				8068
	56030	27000236	27	236	MPY	DELTA				8069
	56031	37000202	37	2	SAR	2				8070
	56032	20001060	20	1060	AUS	PCMD				8071
	56033	31000371	1	56071	LDA	MEA		*****		8072
	56034	04000350	4	350	SUB	HRAD				8073
	56035	06006010	6	56010	TRM	EXIT3		NDA TEST		8074
	56036	01057324	1	57324	LDA	TIMER				8075
	56037	04010235	4	235	SUB	BTIME		FLASH FLT. DIR.		8076
	56040	50007301	10	57301	TRZ*	ETOPTN		BARS IF		8077
	56041	01000235	1	235	LDA	BTIME		BELOW MINIMUM		8078
	56042	12057324	12	57324	STA	TIMLX		DESCENT ALTITUDE		8079
	56043	0101067	1	67	LDA	PCMVLD				8080
	56044	10357010	10	56010	TRZ	EXIT3		*****		8081
	56045	1400067	14	67	EXIT2	STZ		PITCH COMMAND INVALID		8082
	56046	45057301	5	57301	TRA*	ETOPTN				8083
										8084
								FLIGHT DIRECTOR VARIABLES		8085
										8086
								CUTOUT	FT10 750715	8087
								SCALED ALT FOR BANK ANGLE LIMIT	DG	8088
								ENDMOD	FT10 750715	8089
								6 DEG/180 AT 80	DG	8090
								PITCH ANGLE FEEDBACK GAIN AT 83		8091
								DAMPING FACTOR		8092
								G. P. INPUT MULTIPLIER		8093
								HDFE LIMIT COMPUTATION CONSTANT		8094
								GLIDE SLOPE BEAM ERROR GAIN		8095
								0 DEG/180 AT 80		8096
										8097
								COMPUTATIONAL CONSTANT		8098
								PREF COMPUTATIONAL CONSTANT		8099
								HREF COMPUTATIONAL CONSTANT		8100

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX I

C-4000 ASSEMBLY				15 JULY 1975				MSFS C4KR3 FT10 750715				PAGE 217			
FLAGS	LOC	HEX-WORD	I OP	M	LABEL	OPC	OPERAND	COMMENTS				SEQUENCE			
56063	7777640				K10	GEN	-3.815	UPPER LIMIT OF HOCF							
					*****			CUTOUT				FT10 750715 DG			
56064	00001505				K1000	GEN	037.823	SCALED ALT LIM FOR WASHOUT							
					*****			ENDMOD				FT10 750715 DG			
56065	03777777				LIM180	GEN	003777777	180 DEG/180 AT R3							
56066	00000144				LIMLOC	GEN	100.823								
56067	00006472				LIMPR	GEN	.0733333380	PITCH RATE LIMIT .60DEG/180 AT 80							
56070	00000620				LIMY	GEN	400.823	COURSE CUT LIMITER							
56071	00000472				MDA	GEN	250.823	MIN DESCENT ALTITUDE							
56072	01060000				PERIOD	GEN	35.810	NATURAL FREQUENCY							
56073	14040000				TAU1	GEN	3.83	ROLL COMPUTATION - YETA LAG							
56074	46000000				TAU2	GEN	60.86	PSIC WASHOUT TIME CONSTANT							
56075	17000000				TAU3	GEN	1.82	ROLL COMMAND LAG							
56076	06000000				TAU4	GEN	3.84	PITCH COMPUTATION - LAG							
56077	04000000				TAU5	GEN	1.83	PITCH COMMAND LAG							
56100	06000000				TAU6	GEN	1.583	PITCH WASHOUT TIME CONSTANT							
56101	20000000				TF	GEN	16.85	FLARE TO TD TIME SECS							
					*****			ENDMOD				FT7 750115 SL			

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX II

Specifications of EADIFD

Variable Name	Explanation of Mnemonics	Binary Scale	Range or Value
A0	Constant zero	B23	0
A1200	Constant 1200	B23	1200
BINTME	Real time clock	B23	≥ 0 and < 1.80
CAPT	Capture - airplane has captured the ILS beam	Binary	SET or RESET
C180BF	Constant 180 at B15	B15	180.B15
C3B2	Constant 3 at B2	B2	3.B2
C5B15	Constant 5 at B15	B15	5.B15
DEG15	Constant 15 degree	B0	15/180B0
DEG2	Constant 2 degree	B0	2/180B0
DEG20	Constant 20 degree	B0	20/180B0
DEG25	Constant 25 degree	B0	25/180B0
DEG4	Constant 4 degree	B0	4/180B0
DEG6	Constant 6 degree	B0	6/180B0
DELTA	Fast loop frame time	B0	.05B0 or .130
EFORTH	Return address	-	-
FDBK3	Constant feedback gain	B3	.8B3
FLARE	Flare flag	Binary	SET or RESET
FLRALT	Label denoting a location in the routine	-	-
GAMCMD	Flight path angle command	B0	-1.80 to 1.80
GAMMA	Flight path angle	B0	-1.80 to 1.80

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX II

GAM1AC	Intermediate value of flight path angle command	B0	-1.B0 to 1.B0
GRUND	Go around flag (not used at present)	Binary	SET or RESET
GS	Ground speed (in knots)	B15	-.625B15 to .625B15
GSA	Glide slope angle	B0	-1.B0 to 1.B0
GSBSNS	Glide slope beam sensed flag	Binary	SET or RESET
GSDEV	Glide slope deviation	B12	-.00111111B5 to .00111111B5
GSPD	1st pass flag for vertical land mode	Binary	SET or RESET
GSVLD	Glide slope valid flag	Binary	SET or RESET
GJID2D	2D guidance flag	Binary	SET or RESET
GUI3D	3D guidance flag	Binary	SET or RESET
HDCF	Complementary filtered vertical speed	B15	-1.B3 to 1.B8
HRAD	Radar altitude	B23	0 to 1500.B23
HREF	Reference altitude used in computing flare altitude	B15	50.B15 to 1.B0
ILSON	ILS tuned to a valid frequency flag	Binary	SET or RESET
KFLR	Computed flare gain	B8	-1.B15 to 1.B15
KHD	Vertical path velocity error gain	B0	absolute value >.016B0 and <=1.B0
KDA1P	Damping factor	B0	.9B0
K1	Constant value	B11	700.B11

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX II

K2	Constant value	B15	3.B15
K3	Constant value	B0	.03B0
K4	Constant value	B0	0
K5	Constant value	B0	.39B0
K6	Constant value	B0	.653B0
K7	Constant value	B15	2.B15
K8	Constant value	B5	-1.635
K10	Constant value	B15	-3.B15
K1000	Constant scaled value to represent 1000 ft on radar altimeter	B23	837.B23
K1200	Constant scaled value to represent 1200 ft on radar altimeter	B23	925.B23
LATSTR	Lateral steering signal	B0	-1.B0 to 1.B0
LBS	Lateral beam sensed discrete	Binary	SET or RESET
LGAMC	Limited GAMMAC	B0	-10/180B0 to 0
LIMLOC	Gain Programmed Lat. Dev.	B23	100.B23
LIMY	Lateral dev. course cut limit (constant)	B23	400.B23
LIM180	Limit value 180 degree	B3	1.B3
LIMPR	Constant pitch rate limit	B0	.6/130B0
LOCDEV	Localizer deviation	B12	-2/130B5 to 2/180B6
LOCDFD	Land mode 1st pass switch	Binary	SET or RESET
LOCOC	Localizer on course flag	Binary	SET or RESET
LOCSTR	Localizer steering signal	B0	-1.B0 to 1.B0
LOCVLD	Localizer valid flag	Binary	SET or RESET
MJA	Constant minimum descent	B23	250.B23

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX II

	altitude		
MPILS	ILS selected on EADI mode	Binary	SET or RESET
	panel flag		
MSKL15	Constant octal value	-	000077777
PCMD	Pitch command	B0	-6/180B0 to 6/180B0
PCMVLD	Pitch command valid flag	Binary	SET or RESET
PCUWLD	Program control unit mode	Binary	SET or RESET
	word		
PERIOD	Reciprocal of natural	B10	35.B10
	frequency		
PIB2	Pi at B2	B2	3.1415926B2
PITCH	Pitch angle	B0	-1.B0 to 1.B0
PSIE	Heading error	B0	-1.B0 to 1.B0
RCMD	Roll command	B0	-25/180B0 to 25/180B0
RCMVLD	Roll command valid flag	Binary	SET or RESET
RESET	Constant octal value to	-	000000000
	reset binary state variable		
RLFD	Temporary storage location	B0	-1.B0 to 1.B0
	for roll integrator output		
ROLL	Roll angle	B0	-1.B0 to 1.B0
RWYHD3	Runway heading	B0	-1.B0 to 1.B0
SET	Constant octal value to	-	011111111
	set binary state variable		
SRLMT	Limit subroutine	-	-
START	1st pass flag used to	Binary	SET or RESET
	initialize flags, integrators, etc.		

APPENDIX II

TAU1	Constant roll computation lag	B3	3.B3
TAU2	Heading error washout time constant	B6	60.B6
TAU3	Roll command lag	B2	1.B2
TAU4	Pitch computation lag	B4	3.B4
TAU5	Pitch command lag	B3	1.B3
TAU6	Pitch washout time constant	B3	1.5B3
TF	Time from flare to touchdown	B5	16.B5
TIMER	Time counter to flash pitch bar of flight director	B23	0 to 1.B0
TK	Track angle	B0	-1.B0 to 1.B0
TMEPLG	Real time flag	Binary	SET or RESET
TP1	Temporary storage	B3	-1.B3 to 1.B3
TP2	Temporary storage	B3	-1.B3 to 1.B3
TP3	Temporary storage	B15	-1.B13 to 1.B13
TP4	Temporary storage	B0	-1.B0 to 1.B0
TP5	Temporary storage	B0	-1.B0 to 1.B0
TP6	Temporary storage	B0	-1.B0 to 1.B0
TP7	Temporary storage	B3	-1.B3 to 1.B3
T12	Constant value mask to extract bit 12	-	000004000
VERSTR	Vertical steering signal	B15	-1.B3 to 1.B8
VGS	Ground speed in ft/sec	B15	>=100.B15 and < 1.B0
YETA	Lateral deviation from	B23	-1.B0 to 1.B0

APPENDIX II

localizer center in feet

APPENDIX III

- (1) Listing of the Verifier Program
- (2) Theorem Library
- (3) Abbreviation Definitions
- (4) Proofs

```

1  EDUMP = 1
2  ESTLIMIT = 500000
3  EMAXLNGTH = 32700
4  EANCHOR = 1
5  ETRIM = 1
6  DIGIT = '0123456789'
7  OCTAL = '01234567'
8  ALPHA = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
9  VFCD = '*** VERIFICATION CONDITIONS ***'
10 E = 'ER'
11 RD = ('+', '-', '*', '/', '<', '.', '=', '>', ')')
    + '(' | RPOS(0))
12 LOEL = '+-*/<>=.,()'
13 CHAR = DIGIT ALPHA
14 DECIMAL = (SPAN(DIGIT) | '.' SPAN(DIGIT) | SPAN(DIGIT) '.' |
    + SPAN(DIGIT) '.' SPAN(DIGIT))
15 OUTPUT('TITLE'., '(I1,132A1)')
16 DEFINE('CLRTBL()I,Y')
17 DEFINE('CLTIX()I')
18 DEFINE('CNDCT(X0,X1')
19 DEFINE('CONV(X)X0,X1,X2')
20 DEFINE('ENOPCODE(X,Y)X0')
21 DEFINE('FRT(X0)X1,X2,LINE,I')
22 DEFINE('ETSB(X,Y)')
23 DEFINE('EVLIX(X)IX,X0,X1,X2')
24 DEFINE('INST(X,Y)X0')
25 DEFINE('MACR(X)X,MACTXT,MACODE,ARGLIST,MACNAM,ARGS,ARGNO')
26 DEFINE('MACEXP(X)X0,X1,X2,B,C,IX,ARGNO,ARGLIST,MACODE,X')
27 DEFINE('NOSPACE(X)')
28 DEFINE('OUTSTMT(X)W')
29 DEFINE('PUNCHCARDS(X)Y,Z')
30 DEFINE('REPLACE(A,B,C)X,Y,Z')
31 DEFINE('RSTR(X)I,X0')
32 DEFINE('SCALE(X)X1,X2,X3,X4,X5')
33 DEFINE('SUB(X)X0,X1,X2,X3,IX')
34 DEFINE('SYXLBL(X)')
35 MACR()
36 MCEPLV = TRIM(INPUT)
37 OPCODE = TABLE(50)
38 IX = 1
39 X = 'LDALQSTASTQSTZANAORAADDSUBADS'
40 ENOPCODE(X,3)
41 X = 'SBSMPYINCDECTNETRATRMTRPTRZTNZ'
42 ENOPCODE(X,3)
43 X = 'IASSALSARABSCMAHLTSEXXAQDQIV8TD'
44 ENOPCODE(X,3)
45 X = 'OTBLAQSAQDPAOPSSLLSLRBSSGENEQUENDREM'
46 ENOPCODE(X,3)
47 X = 'ADDSUBOADSOSBSOINCODECOPDAQDPSO'
48 ENOPCODE(X,4)
49 SYMBOL = TABLE(50,10) : (BBD)
50 BBD CLRTBL()
51 BBD ASST = ARRAY('50')
52 CPGN = ARRAY('30,5')
53 TEXT = ARRAY('300,8')
54 SYMTBL = ARRAY('160,6') : (BBA)

```

```

*****
*   DEFINE('CLTIX()I')
*****
*   COLLECT INDEX OF VARIABLES
55 CLTIX CLTIX =
56     I = 0
57 CX1 I = LT(I,SYMX) I + 1 :F(RETURN)
58 CLTIX = CLTIX ;' SYMTBL<1,4> :{CX1}
*****
*   DEFINE('CLRTBL()I,Y')
*****
59 CLRTBL SYMBOL = CONVERT(SYMBOL,'ARRAY') :F(RETURN)
60     Y = PROTOTYPE(SYMBOL)
61     Y BREAK('') . Y
62     I = 1
63 CL1 SYMBOL<1,2> =
64     I = LT(I,Y) I + 1 :S(CL1)
65     SYMBOL = CONVERT(SYMBOL,'TABLE') :{RETURN}
*****
*   DEFINE('CNDC()X0,X1')
*****
66 CNDC  &ANCHOR = 0
67     X0 = CON
68     CON =
69 CNDC1  X0 DIFFER(X0) BAL . X1 ('.' | RPOS(0)) = :F(CNDC3)
70     HYP ('.' | POS(0)) X1 ('.' | RPOS(0)) :F(CNDC2)
71     X1 POS(0) BAL RPOS(0) :S(CNDC1)
72 CNDC2  CON = CON . X1 :{CNDC1}
73 CNDC3  &ANCHOR = 1
74     CON . = :{RETURN}
*****
*   DEFINE('CONV(X)X0,X1,X2')
*****
*   CONVERT OPERAND INTO ITS DECIMAL EQUIVALENT
*   0 <OCTAL INTEGER>
*   <GENERAL DECIMAL> | <GENERAL DECIMAL> B <INTEGER>
*   MIXED : ONE TAG AND ONE DECIMAL DECREMENT OR INCREMENT
*   SYMBOLIC
75 CONV X (('+ | '- | '') . X0 '0' SPAN(OCTAL) . X1 POS(0)) :F(CV2)
*   OCTAL NUMBER
76     CONV = LT(X1,8) X1 :F(CV1)
77     CONV = IDENT(X0,'-') '-' CONV :{CV1B}
78 CV1  X1 DIFFER(X1) LEN(1) . X2 = :F(CV1A)
79     CONV = CONV . 8 + X2 :{CV1}
80 CV1A  CONV = DIFFER(X0,'+') X0 CONV
81 CV1B  CONV = CONV /2**23 :{RETURN}
*   SCALED DECIMAL NUMBERS
82 CV2  X (('+ | '- | POS(0)) . X0 DECIMAL . CONV RPOS(0)) :F(CV2A)
83     CONV = CONV /2**23 :{RETURN}
84 CV2A  X (('+ | '- | '') . X0 DECIMAL . CONV '8' (('+ | '- | '') . X1
+       SPAN(DIGIT) . X2 RPOS(0)) :F(CV3)
85     CONV = IDENT(X1,'-') CONV *2** X2 :{RETURN}
86     CONV = CONV /2** X2 :{RETURN}
*   MIXED EXPRESSION
87 CV3  X BREAK('+-') . X0 LEN(1) . X1 SPAN(DIGIT) . X2 :F(CV4)
88     SYXLBL(X0) :F(S(E'CV3A'))

```

```

89      IX = SYMBOL<X0>
90      IDENT(IX) :S($E 'CV3B')
91      IY = SYMTBL<IX,2>
92      IDENT(IY) :S($E 'CV3B')
93      IY = EVAL(IY ' ' X1 ' ' X2)
94      IX = I
95 CV3A IDENT(IY.SYMTBL<IX,2>) :S(CV3B)
96      IX = LT(IX.SYMX) IX + 1 :S(CV3A)F($E 'CV3A')
97 CV3B SYMTBL<IY,1> 'C;' REM . CONV :S(RETURN)
* SYMBOLIC
98 CV4 SYXLBL(X) :F($E 'CV4A')
99      IX = SYMBOL<X>
100     IDENT(IX) :S($E 'CV4B')
101     SYMTBL<IX,1> 'C;' REM . CONV :S(RETURN)F($E 'CV4C')
*****
* DEFINE('ENOPCODE(X,Y)X0')
*****
* ENTER OPCODES INTO OPCODE TABLE
102 ENOPCODE X LEN(Y) . X0 = :F(RETURN)
103     OPCODE<X0> = IX
104     IX = IX + 1 :S(ENOPCODE)
*****
* DEFINE('ETSD(X,Y)')
*****
* ENTER SYMBOL INTO SYMBOL TABLE
105 ETSD SYMX = SYMX + 1
106     SYMBOL<X> = SYMX
107     SYMTBL<SYMX,1> = Y
108     SYMTBL<SYMX,3> = IDENT(Y.'V') X :S(RETURN)
109     SYMTBL<SYMX,3> = IDENT(Y.'A') X :S(RETURN)
*****
* DEFINE('EVLIX(X)IX.X0,X1,X2')
*****
* EVALUATE THE EXPRESSIONS; X IS AN EXPRESSION OF LABELS(LOCATIONS)
* IY IS A GLOBAL VARIABLE WHICH CONTAINS THE CURRENT LINE COUNT
110 EVLIX X BREAK('+-') . X0 LEN(1) . X1 SPAN(DIGIT) . X2 RPOS(0) :F(EX1)
111     IDENT(X0.'**') :F(EX0)
112     IX = EVAL(TEXT<IY,8> ' ' X1 ' ' X2)
113     EVLIX = IY
114 EX0A IDENT(TEXT<EVLIX,8>,IX) :S(RETURN)
115     EVLIX = LT(EVLIX,LIC) EVLIX + 1 :S(EX0A)F($E 'EVLIX2')
116 EX0 IX = SYMBOL<X0>
117     IDENT(IX) :S($E 'EVLIX1')
118     IDENT(SYMTBL<IX,1>,'L') :F($E 'EVLIX1')
119     IX = EVAL(SYMTBL<IX,3> ' ' X1 ' ' X2)
120     EVLIX = IY :S(EX0A)
121 EX1 SYXLBL(X) :F($E 'EVLIX2')
122     IX = SYMBOL<X>
123     EVLIX = DIFFER(IX) IDENT(SYMTBL<IX,1>,'L') SYMTBL<IX,3>
* :S(RETURN)F($E 'EVLIX1')
*****
* DEFINE('FRT(X0)X1,X2,LINE,1')
*****
124 FRT LINE =
125     I = 0
126 FRT0 X0,DIFFER(X0) BAL . X1 ('.' | RPOS(0)) = :F(FRT7)

```

```

127 FRT1 N = SIZE(X1) / 22 + 1
128 EQ(I,5) GT(N,1) :S(FRT5)
129 FRT2 X1 LEN(22) . X2 = :F(FRT3)
130 LINE = LINE RPAD(X2,22)
131 I = I + 1
132 EQ(I,6) :S(FRT5)F(FRT2)
133 FRT3 LINE = LINE RPAD(X1,22)
134 I = I + 1
135 FRT4 EQ(I,6) :F(FRT0)
136 OUTPUT = LINE
137 I = 0
138 LINE = : (FRT0)
139 FRT5 OUTPUT = LINE
140 FRT6 X1 LEN(132) . OUTPUT = :S(FRT6)
141 IDENT(X1) :S(FRT)
142 I = SIZE(X1) / 22 + 1
143 LINE = RPAD(X1,I * 22) : (FRT4)
144 FRT7 OUTPUT = LINE : (RETURN)
*****
* DEFINE('INST(X,Y)X0')
*****
* REPLACE 'S IN X BY Y
145 INST X BREAK('S') . X0 LEN(1) = X0 Y :S(INST)
146 INST = X : (RETURN)
*****
* DEFINE('MACEXP(X)X0,X1,X2,B,C,IX,ARGNO,ARGLIST,MACODE,A')
*****
* MACRO EXPANSION SUBROUTINE ; EXPAND MACROS IN X BY ITS DEFINITION IF ANY.
*****
147 MACEXP MACEXP =
148 MACEXP = EQ(MCEPLV,0) X :S(RETURN)
149 A = 1
150 MP1 X (SPAN(LDEL) | NULL) . X0 SPAN(CHAR 'S') . X1 RD . X2 = :F(MP6)
151 IX = MACRO(X1)
152 IDENT(IX) :S(MP5)
153 ARGNO = MACTBL<IX,1>
154 ARGLIST = MACTBL<IX,2>
155 MACODE = MACTBL<IX,3>
156 EQ(ARGNO,0) :S(MP4)
157 MP2 EQ(ARGNO,1) :S(MP3)
158 X BAL . C 'S' = :F($E 'MP1'))
159 DIFFER(C) :F($E 'MP3'))
160 C = MACEXP(C)
161 ARGLIST BAL . B 'S' = :F($E 'MP2'))
162 MACODE = REPLACE(MACODE,B,C)
163 ARGNO = ARGNO - 1 : (MP2)
164 MP3 X BAL . C 'S' = :F($E 'MP1'))
165 DIFFER(C) :F($E 'MP3'))
166 C = MACEXP(C)
167 B = ARGLIST
168 MACODE = REPLACE(MACODE,B,C)
169 MACEXP = MACEXP X0 MACODE : (MP1)
170 MP4 MACEXP = MACEXP MACEXP(MACODE) : (MP1)
171 MP5 MACEXP = MACEXP X0 X1 X2 : (MP1)
172 MP6 A = LT(A,MCEPLV) A + 1 :F(RETURN)
173 X = MACEXP : (MP1)

```



```

*****
*   DEFINE('MACR(X,MACTXT,MACODE,ARGLIST,MACNAM,ARGS,ARGNO')
*****
*   MACOR IS THE TABLE WHICH CONTAINS MACRO NAMES.
*   MACTBL IS THE ARRAY WHICH CONTAINS # OF ARGUMENTS, MACRO ARGUMENT LIS .T. AND
*   MACRO DEFINITION
174 MACR MACX = 0
175 TITLE =
176 OUTPUT = '--- USER DEFINED MACROS ---'
177 OUTPUT =
178 MACRO = TABLE(30,30)
179 MACTBL = ARRAY('130,4')
180 TYPE = 'P'
181 MO0 MACTXT =
182 MO1 X = TRIM(INPUT)
183 IDENT(X,'222222222') :S(RETURN)
184 TYPE = IDENT(X,'111111111') 'E' :S(MO1)
185 OUTPUT = X
186 IDENT(TYPE,'E') :S(MO1)
187 X LEN(79) . X 'C' :F(MO2)
188 MACTXT = MACTXT X :S(MO1)
189 MO2 MACTXT = MACTXT X
190 MACTXT BAL . X '=' REM . MACODE :F(S(E 'MO1'))
191 ARGNO = 0
192 ARGLIST =
193 X BREAK('(') . MACNAM LEN(1) RTAB(1) . ARGS ')' :S(MO3)
194 X (ANY(Alpha) SPAN(CHAR) RPOS(0)) . MACNAM :S(MO4)F(S(E 'MO1'))
195 MO3 ARGS DIFFER(ARGS) BAL . X ('.' | RPOS(0)) = :F(MO4)
196 ARGNO = ARGNO + 1
197 MACODE = REPLACE(MACODE,X,'S' X)
198 ARGLIST = DIFFER(ARGLIST) ARGLIST 'S' X :S(MO3)
199 ARGLIST = 'S' X :S(MO3)
200 MO4 MACX = MACX + 1
201 MACRO(MACNAM) = MACX
202 MACTBL(MACX,1) = ARGNO
203 MACTBL(MACX,2) = ARGLIST
204 MACTBL(MACX,4) = TYPE
205 MACTBL(MACX,3) = MACODE :S(MO0)
*****
*   DEFINE('NOSPACE(X)')
*****
206 NOSPACE SANCHOR = 0
207 NE1 X ' ' = :S(NE1)
208 NOSPACE = X
209 SANCHOR = 1 :S(RETURN)
*****
*   DEFINE('OUTSTMT(X)W')
*****
210 OUTSTMT W = SIZE(X) / 20 + 1
211 OUTPUT = GT(STMTNO + W,6) STMTS :F(OT2)
212 STMTNO = W
213 STMTS = RPAD(X,W * 20) :S(RETURN)
214 OT2 STMTNO = STMTNO + W
215 STMTNO = EQ(STMTNO,6) 0 :F(OT1)
216 OUTPUT = STMTS X
217 STMTS = :S(RETURN)

```

```

218 OT1 STMTS = STMTS RPAD(X,20 * W) : (RETURN)
*****
* DEFINE('PUNCHCARDS(X)Y,Z')
*****
219 PUNCHCARDS
220 CANCHOR = 0
221 PS1 X *S' = 'C' :S(PS1)
222 CANCHOR = 1
223 PS2 PUNCH = LE(SIZE(X),79) X :S(RETURN)
224 X LEN(79) . Z =
225 PUNCH = Z *C' : (PS2)
*****
* DEFINE('REPLACE(A,B,C)X,Y,Z')
*****
* REPLACE ALL B'S IN A BY C
226 REPLACE REPLACE =
227 RE1 A (SPAN(LOEL) | POS(0)) . X SPAN(CHAR 'S') . Y RD . Z = :F(RE3)
228 IDENT(B,Y) :F(RE2)
229 REPLACE = REPLACE X C Z : (RE1)
230 RE2 REPLACE = REPLACE X Y Z : (RE1)
231 RE3 REPLACE = REPLACE A : (RETURN)
*****
* DEFINE('RSTRIX(X)I,X0')
*****
* RESTORE INDEX OF VARIABLES
232 RSTRIX I = 1
233 RX1 X *I' BREAK('I') . X0 = :F(RETURN)
234 SYMTBL<I,4> = X0
235 I = I + 1 : (RX1)
*****
* DEFINE('SCALE(X).X1,X2,X3,X4,X5')
*****
236 SCALE
237 SE1 X DIFFER(X) *(' BREAK('I') . X2 *I' BREAK('I') . X3 *I'
+ BREAK('I') . X4 *I' BREAK('I')) . X6 *I' ('I' | RPOS(0)) =
+ :F(RETURN)
238 X5 = SYMBOL<X2>
239 IDENT(X5) :F(SE2)
240 ETSB(X2,X3)
241 SYMTBL<SYMX,6> = X6
242 SYMTBL<SYMX,5> = X4 : (SE1)
243 SE2 IDENT(SYMTBL<X5,1>,X3) IDENT(SYMTBL<X5,5>,X4) :S(SE1)F($E 'SE'))
*****
* DEFINE('SUB(X)X0,X1,X2,X3,IX')
*****
* APPEND INDEX TO VARIABLES IN STRING X
244 SUB SUB =
245 SB0 IDENT(X) :S(RETURN)
246 X (SPAN(LOEL) | POS(0)) . X0 SPAN(CHAR 'S') . X1 RD . X2 =
+ :F(SB4)
247 X1 *S' :S(SB2)
248 SYXLBL(X1) :F(SB1)
249 SB2 IX = SYMBOL<X1>
250 X3 = DIFFER(IX) SYMTBL<IX,1> :F(SB1)
251 X3 *C' (('I' | 'I') DECIMAL ) . X1 RPOS(0) :S(SB1)
252 EQ(SYMTBL<IX,4>,0) :S(SB1)

```

```

253      X3 ('V' | 'A') :S(SB3)
254 SB1  SUB = SUB X0 X1 X2 :S(SB0)
255 SB3  SUB = SUB X0 X1 SYMTBL<X,4> X2 :S(SB0)
256 SB4  SUB = SUB X :S(RETURN)
*****
*      DEFINE('SYXLBL(X)')
*****
* CHECK WHETHER X IS A LABEL OR NOT
257 SYXLBL X POS(0) SPAN(DIGIT) RPOS(0) :S(FRETURN)
258      X POS(0) SPAN(CHAR) RPOS(0) :F(FRETURN)
259      X POS(0) 'D' :F(FRETURN) S(FRETURN)
260
*****
* READ IN STATEMENTS
*****
261
262 BBA  SYMX = 0
263     ASTX = 0
264     CPGNTX = 0
265     RETADD = 'UNDEFINED.'
266     EXIT =
267     SPEC =
268     TITLE = '--- SPECIFICATION ---'
269 BBA00 LINE = INPUT
270     IDENT(LINE,'7777777777') :S(BBA01)
271     OUTPUT = LINE
272     LINE SPAN('**') SPAN(' ') =
273     SPEC = SPEC NOSPACE(LINE) :S(BBA00)
274 BBA01 OUTPUT =
275     OUTPUT =
276     TITLE = '--- SCALING ---'
277 BBA0 LINE = NOSPACE(INPUT)
278     IDENT(LINE,'8888888888') :S(BBA1)
279     OUTPUT = LINE
280     SCALE(LINE) :S(BBA0)
281 BBA1  ETSB('SAC','V') ; ETSB('SRA','V')
282     ETSB('SMQ','V') ; ETSB('SAQ','V')
283
284     SAC =
285     SMQ =
286     LIC = 0 ; LOC = 0
287
288     TITLE =
289     OUTPUT = '--- LISTING OF THE PROGRAM TO BE VERIFIED ---'
290
291     OUTPUT =
292 BBA0  LIC = LIC + 1
293 BBA1  INOREF = 0
294     LINE = TRIM(INPUT)
295     IDENT(LINE,'0000000000') :S(END)
296     IDENT(LINE,'9999999999') :S(EE0)
297     OUTPUT = LINE
298     LINE SPAN('**') SPAN(' ') = :F(BBA)
299     LINE = NOSPACE(LINE)
300     IDENT(LINE) :S(BB1)
301 BBA2  LINE1 = TRIM(INPUT)
302     OUTPUT = LINE1
303     LINE1 SPAN('**') SPAN(' ') = :F(BB3)
304     IDENT(LINE1) :S(BB2)

```

```

305     LINE = LINE NOSPACE(LINE1) :S(BB2)
306 BB3  ASTX = ASTX + 1
307     ASST<ASTX> = LIC
308     TEXT<LIC.1> = LINE
309     LINE = LINE1
310     IDENT(LINE,'9999999999') :S(EE0)
311 BB4  LINE GT(SIZE(LINE),25) LEN(25) . LINE :F(BB5)
312     LINE = TRIM(LINE)
313 BB5  LINE LEN(6) . LABEL LEN(1) (BREAK(' ') . OP SPAN(' ') REM . OPND
+      | REM . OP NULL . OPND) :F($E 'BB1')
314     LABEL = TRIM(LABEL)
315     IDENT(LABEL) :S(BB8)
316     SYXLBL(LABEL) :F($E 'BB3')
317 BB8  OP LEN(3) . OP '** RPOS(0) :F(BB6)
318     INDREF = 1
319     IOP = OPCODE<OP>
320     LE(15,IOP) LE(IOP,21) :S(BC0)F($E 'BB1')
321 BB6  IOP = OPCODE<OP>
322     IDENT(IOP) :S($E 'BB2')
323

```

```

*****
* STATEMENT DECODING
*****

```

```

324
325 BC0  LINE LEN(7) =
326 REMA  EQ(IOP,42) :S(BB1)
327 BSS  EQ(IOP,38) :F(GEN)
328     IDENT(LABEL) :S($E 'BSS0')
329     IX = SYMBOL<LABEL>
330     IDENT(IX) :S(BSS1)
331     SYMTBL<IX,1> ('U' | 'V' | 'VS' | 'V') :S(BSS2)F($E 'BSS1')
332 BSS1  SYMX = SYMX + 1
333     SYMBOL<LABEL> = SYMX
334     IX = SYMX
335 BSS2  TEMP = CONV(OPND) :F($E 'BSS2')
336     SYMTBL<IX,1> = IDENT(TEMP,'1') *V :S(BSS3)
337     SYMTBL<IX,1> = 'A:' TEMP
338 BSS3  SYMTBL<IX,2> = LOC
339     SYMTBL<IX,3> = LABEL
340     LOC = LOC + TEMP :S(BB1)
+      * SUPPOSE THAT GENERATED DATA CANNOT BE CHANGED DURING EXECUTION
341 GEN  EQ(IOP,39) :F(EQU)
342     IDENT(LABEL) :S($E 'GEN0')
343     IX = SYMBOL<LABEL>
344     IDENT(IX) :F(GN1)
345     SYMX = SYMX + 1
346     SYMBOL<LABEL> = SYMX
347     IX = SYMX
348 GN1  X2 =
349     OPND BREAK('B') *B REM . X2
350     SYMTBL<IX,1> = 'C:' OPND :F($E 'GEN2')
351     SYMTBL<IX,2> = LOC
352     SYMTBL<IX,3> = LABEL
353     SYMTBL<IX,5> = IDENT(X2) *B :S(GN2)
354     SYMTBL<IX,5> = X2
355 GN2  LOC = LOC + 1 :S(BB1)

```

```

356 EQU EQ(IOP,40) :F(ENDS)
357 OPND = CONV(OPND) :F(EU2)
* EQUATE CONSTANT TO CONSTANT VARIABLE
358 SYXLBL(LABEL) :F($E 'EQU1')
359 IX = SYMBOL<LABEL>
360 SYMX = DIFFER(IX) SYMX + 1 :F(EU3)
361 IX = SYMX
362 EU3 SYMTBL<IX,1> = 'C' OPND : (BB1)
* EQUATE LABEL TO ANOTHER LABEL
363 EU2 SYXLBL(LABEL) :F($E 'EQU1')
364 IX = SYMBOL<OPND>
365 IDENT(IX) :F($E 'EQU2')
366 SYMX = IDENT(SYMBOL<LABEL>) SYMX + 1 :F(EU1)
367 SYMBOL<LABEL> = SYMX
368 IX = SYMX
369 EU1 SYMTBL<IX,1> = 'L'
370 IW = SYMBOL<OPND>
371 IDENT(IW) :F($E 'EQU2')
372 SYMTBL<IX,1> = 'L'
373 SYMTBL<IX,2> = SYMTBL<IW,2>
374 SYMTBL<IX,3> = SYMTBL<IW,3> : (BB1)
375 ENDS EQ(IOP,41) :F(BC1)
376 TEXT<LIC,2> = LINE
377 TEXT<LIC,5> = 'X' : (BB0)
378 BC1 IDENT(LABEL) :S(BC3)
379 IX = SYMBOL<LABEL>
380 IDENT(IX) :S(BC2)
381 SYMTBL<IX,1> 'U' | 'L' :F($E 'BC1')S(BC2A)
382 BC2 SYMX = SYMX + 1
383 SYMBOL<LABEL> = SYMX
384 IX = SYMX
385 BC2A SYMTBL<IX,1> = 'L'
386 SYMTBL<IX,2> = LOC
387 SYMTBL<IX,3> = LIC
388 BC3 LOC = LOC + 1
389 GE(IOP,15) LE(IOP,20) :F(BC8)
390 EQ(INDREF,1) :F(BC3A)
391 SYXLBL(OPND) :F($E 'BC3A')
392 IX = SYMBOL<OPND>
393 IDENT(SYMTBL<IX,1>,'V$') :S(BC7)F($E 'BC3A')
394 BC3A OPND BREAK('+-') * TAG LEN(1) SPAN(DIGIT) * DEV :F(BC5)
* MIXED OPERAND
395 IDENT(TAG,'*') :S(BC7)
396 SYXLBL(TAG) :F($E 'BC2')
397 IX = SYMBOL<TAG>
398 IDENT(IX) :F(BC4)
399 SYMX = SYMX + 1
400 SYMTBL<SYMX,1> = 'L' : (BC7)
401 BC4 SYMTBL<IX,1> ('L' | 'U') RPOS(0) = 'L' :F($E 'BC3')S(BC7)
402 BC5 SYXLBL(OPND) :F($E 'BC5')
403 IX = SYMBOL<OPND>
404 IDENT(IX) :F(BC6)
405 SYMX = SYMX + 1
406 SYMBOL<OPND> = SYMX
407 SYMTBL<SYMX,1> = 'L' : (BC7)
408 BC6 SYMTBL<IX,1> ('L' | 'U') * X RPOS(0) = 'L' :F($E 'BC6')

```

```

409 BC7 TEXT<LIC,2> = LINE
410 TEXT<LIC,4> = EQ(INDREF,1) '**
411 TEXT<LIC,8> = LOC :($('CC' IOP))
412 CC15 TEXT<LIC,6> = '** : (BB0A)
413 CC16 TEXT<LIC,5> = IDENT(OPND,'SRLMT') *SAC=SRLMT(SAC,$MQ)'
+ :S(BB0A)
414 TEXT<LIC,5> = '**
415 TEXT<LIC,6> = '** : (BB0A)
416 CC17 TEXT<LIC,5> = '**
417 TEXT<LIC,6> = 'SAC.LT.0' : (BB0A)
418 CC18 TEXT<LIC,5> = '**
419 TEXT<LIC,6> = 'SAC.GE.0' : (BB0A)
420 CC19 TEXT<LIC,5> = '**
421 TEXT<LIC,6> = 'SAC.EQ.0' : (BB0A)
422 CC20 TEXT<LIC,5> = '**
423 TEXT<LIC,6> = 'SAC.NE.0'
424 BB0A TEXT<LIC,7> = OPND : (BB0)
425 BC8 TEXT<LIC,2> = LINE
426 TEXT<LIC,4> = OPND
427 TEXT<LIC,8> = LOC : ($('DD' IOP))
428 DD1 TEXT<LIC,5> = 'SAC=#' : (BB0)
429 DD2 TEXT<LIC,5> = '$MQ=#' : (BB0)
430 DD3 TEXT<LIC,5> = '#=SAC' : (BB0)
431 DD4 TEXT<LIC,5> = '#=$MQ' : (BB0)
432 DD5 TEXT<LIC,5> = '#=0' : (BB0)
433 DD6 TEXT<LIC,5> = 'SAC=AND(SAC,#)' : (BB0)
434 DD7 TEXT<LIC,5> = 'SAC=OR(SAC,#)' : (BB0)
435 DD8 TEXT<LIC,5> = 'SAC=ADD(SAC,#)' : (BB0)
436 DD9 TEXT<LIC,5> = 'SAC=SUB(SAC,#)' : (BB0)
437 DD10 TEXT<LIC,5> = 'SAC=ADS(SAC,#),#=$AC' : (BB0)
438 DD11 TEXT<LIC,5> = 'SAC=SBS(SAC,#),#=$AC' : (BB0)
439 DD12 TEXT<LIC,5> = '($MQ,SAC)=(MULTM(SAC,#),MULT(SAC,#))' : (BB0)
440 DD13 IX = SYMBOL<OPND>
441 IDENT(SYMBOL<IX,6>,'I') : F($('E 'DD13'))
442 TEXT<LIC,3> = 'R(#+1)'
443 TEXT<LIC,5> = '#=#+1' : (BB0)
444 DD14 IX = SYMBOL<OPND>
445 IDENT(SYMBOL<IX,6>,'I') : F($('E 'DD14'))
446 TEXT<LIC,3> = 'R(#-1)'
447 TEXT<LIC,5> = '#=#-1' : (BB0)
448 DD21 IX = SYMBOL<OPND>
449 IDENT(SYMBOL<IX,6>,'I') : F($('E 'DD21'))
450 TEXT<LIC,6> = 'R(#+1)'
451 TEXT<LIC,4> = OPND
452 TEXT<LIC,6> = '#=0'
453 TEXT<LIC,7> = '**2'
454 TEXT<LIC,8> = LOC : (BB0)
455 DD22 OPND = IDENT(OPND) '0' : S(DD22A)
456 OPND SPAN(DIGIT) RPOS(0) : F($('E 'DD22'))
457 DD22A TEXT<LIC,5> = 'SAC=SAC*2**' OPND
458 TEXT<LIC,3> = 'R($SAC*2** OPND)' : (BB0)
459 DD23 OPND = IDENT(OPND) '0' : S(DD23A)
460 OPND SPAN(DIGIT) RPOS(0) : F($('E 'DD23'))
461 DD23A TEXT<LIC,5> = 'SAC=SAC/2**' OPND : (BB0)
462 DD24 TEXT<LIC,5> = 'SAC=ABS(SAC)' : (BB0)
463 DD25 TEXT<LIC,5> = 'SAC=-SAC' : (BB0)

```

```

464 DD26 TEXT<LIC,5> = 'X'           : (BB0)
465 DD27 TEXT<LIC,5> = 'N=SRA'
466 SYXLBL(OPND) : F($E 'DD27')
467 IX = SYMBOL<OPND>
468 IDENT(IX) : S(DD27A)
469 SYMTBL<IX,1> ('U' | 'V') : S(DD27B)F($E 'DD27A')
470 DD27A SYMX = SYMX + 1
471 SYMBOL<OPND> = SYMX
472 IX = SYMX
473 DD27B SYMTBL<IX,1> = 'VS'
474 EXIT = IDENT(EXIT) OPND : F($E 'DD27B')
475 SYMTBL<IX,3> = OPND : (BB0)
476 DD28 TEXT<LIC,5> = '($AC,$MQ)=$MQ,$AC)' : (BB0)
477 DD29 TEXT<LIC,5> = '$AC=$AC/)'
478 TEXT<LIC,3> = 'ABS($AC).LT.ABS(N)' : (BB0)
479 DD30 TEXT<LIC,3> = 'BCT(N)'
480 TEXT<LIC,5> = '$AC=BTDL$AC)' : (BB0)
481 DD31 TEXT<LIC,3> = 'BTC(N)'
482 TEXT<LIC,5> = '$AC=DTB($AC)' : (BB0)
483 DD32 TEXT<LIC,5> = '($MQ,$AC)=$N,NN)' : (BB0)
484 DD33 TEXT<LIC,5> = '($N,NN)=$MQ,$AC)' : (BB0)
485 DD34 TEXT<LIC,5> = '($MQ,$AC)=DPA(($MQ,$AC),($N,NN))' : (BB0)
486 DD35 TEXT<LIC,5> = '($MQ,$AC)=DPS(($MQ,$AC),($N,NN))' : (BB0)
487 DD36 OPND = IDENT(OPND) '0' : S(DD36A)
488 OPND SPAN(DIGIT) RPOS(0) : F($E 'DD36')
489 DD36A TEXT<LIC,5> = '$AQ=($AC,$MQ), $AQ=$AQ*2**' OPND
+ '$AC=AR($AQ), $MQ=MR($AQ)'
490 TEXT<LIC,3> = 'DIR($AQ*2**' CPND '))' : (BB0)
491 DD37 OPND = IDENT(OPND) '0' : S(DD37A)
492 OPND SPAN(DIGIT) RPOS(0) : F($E 'DD37')
493 DD37A TEXT<LIC,5> = '$AQ=($AC,$MQ), $AQ=$AQ*2**' OPND
+ '$AC=AR($AQ), $MQ=MR($AQ)' : (BB0)
*****
* CONTROL PATH GENERATION AND VERIFICATION CONDITION GENERATION
*****
494
495 EE0 II = 0
496 CPGNTX = 0
497 TITLE =
498 OUTPUT = 'ABBREVIATION : SPEC IS'
499 OUTPUT = SPEC
500 TITLE =
501 JJ = 0
* II = INDEX OF INITIAL ASSERTION FOR THE CURRENT CONTROL PATH
* JJ = INDEX OF THE CURRENT CONTROL PATH
502 EE1 II = LT(II,ASTX - 1) II + 1 : F(BBB)
503 IY = ASST(II)
504 IDENT(TEXT<IY,4>,'**') : F(EE2)
505 IDENT(TEXT<IY,7>,EXIT) : S(EE1)F($E 'EE0')
506 EE2 JJ = JJ + 1
507 CPTH =
508 OUTPUT =
509 OUTPUT = '--- CONTROL PATH ' JJ ' ---'
510 OUTPUT =
511 CON =
512 STMTND = 0

```



```

513 STMTS =
514 K = 1
515 EE3 SYMTBL<K,4> = 0
516 K = LT(K,SYMX) K + 1 :S(EE3)
517 IY = ASST<II>
518 INASST = TEXT<IY,1>
519 STMT = '** INASST **'
520 CPTH = STMT
521 OUTPUT = STMT
522 HYP = SUB(INASST)
523 IDENT(TEXT<IY,5>,'X') :F(EE4A)
524 IDENT(TEXT<IY,1>) :S(S(E 'EE0'))
525 EE3A CON = DIFFER(CON) CON ' SUB(TEXT<IY,1>) :S(EE3A2)
526 CON = SUB(TEXT<IY,1>)
527 EE3A2 OUTPUT = DIFFER(STMTS) STMTS
528 OUTPUT = '** TEXT<IY,1> **'
529 OUTPUT =
530 OUTPUT = VFCD
531 OUTPUT =
532 PUNCH = '*** FLIGHT DIRECTOR ROUTINE: CONTROL PATH * JJ
533 HYP = MACEXP(HYP)
534 FRT(HYP)
535 PUNCHCARDS(HYP)
536 PUNCH = '**'
537 OUTPUT = '.IMPLIES.'
538 CON = MACEXP(CON)
539 CNDC()
540 FRT(CON)
541 PUNCHCARDS(CON)
542 PUNCH = '**'
543 PUNCH = '**'
544 PUNCH = '888888888888'
545 EE3C1 EQ(CPGNTX,0) :S(EE1)F(EE5)
546 EE4 IDENT(TEXT<IY,1>) :F(EE3A3)
547 EE4A IDENT(TEXT<IY,5>) :S(EE4B)
548 IDENT(TEXT<IY,5>,'**') :F(EE7)
549 EE4B IDENT(TEXT<IY,6>) :F(EE6A)
550 IY = IY + 1 :S(EE4)
551 EE3A3 IDENT(TEXT<IY,4>,'**') :F(EE3A)
552 IDENT(TEXT<IY,6>,'**') :S(EE3A)
553 CPGNTX = CPGNTX + 1
554 CPGN<CPGNTX,1> = '.NOT.(' TEXT<IY,6> *)'
555 CPGN<CPGNTX,2> = IY + 1
556 CPGN<CPGNTX,3> = HYP '**' CON
557 CPGN<CPGNTX,4> = CLTIX()
558 STMT = '(' TEXT<IY,6> *)'
559 OUTSTMT(STMT)
560 CPTH = CPTH '**' STMT
561 CPGN<CPGNTX,5> = CPTH
562 HYP = HYP '**' SUB(TEXT<IY,6>) :S(EE3A)
* POP UP CPGN
563 EE5 TEMP = CPGN<CPGNTX,3>
564 TEMP BREAK('**') . HYP LEN(1) REM . CON
565 RSTRIX(CPGN<CPGNTX,4>)
566 CPTH = CPGN<CPGNTX,5>
567 JJ = JJ + 1

```

```

568     OUTPUT =
569     OUTPUT = '--- CONTROL PATH ' JJ ' ---'
570     OUTPUT =
571     X = 0
572     STMTNO = 0
573     STMTS =
574     CPTH (BREAK(')') . OUTPUT ')') . Y
575     X = SIZE(Y)
576 EE5B  CPTH (TAB(X) BREAK(')') . STMT ')') . Y :F(EE5C)
577     OUTSTMT(STMT)
578     X = SIZE(Y) :((EE5B)
579 EE5C  Y RTAB(1) . CPTH =
580     STMT = '(' CPGN<CPGNTX,1> ')'
581     OUTSTMT(STMT)
582     CPTH = CPTH ')' STMT
583     HYP = HYP DIFFER(CPGN<CPGNTX,1>,'') ' SUB(CPGN<CPGNTX,1>)
584     IY = CPGN<CPGNTX,2>
585     CPGNTX = CPGNTX - 1
586     IDENT(Y,'') :S(EE3A)
587     RETADD = IY + 1 :((EE4)
588 * PUSH DOWN CPGN
589 EE6A  IY = IDENT(TEXT<IY,6>,'') EVLIX(TEXT<IY,7>) :S(EE4)
590     CPGNTX = CPGNTX + 1
591     CPGN<CPGNTX,1> = TEXT<IY,6>
592     IDENT(TEXT<IY,7>,EXIT) :F(EE6B)
593     CPGN<CPGNTX,2> = IDENT(TEXT<IY,4>,'') ' :S(EE6C)
594 EE6B  CPGN<CPGNTX,2> = EVLIX(TEXT<IY,7>)
595 EE6C  CPGN<CPGNTX,3> = HYP ' CON
596     CPGN<CPGNTX,4> = CLTIX()
597     STMT = '.NOT.(' TEXT<IY,6> ')'
598     OUTSTMT(STMT)
599     CPTH = CPTH ' STMT
600     CPGN<CPGNTX,5> = CPTH
601     HYP = HYP ' .NOT.(' SUB(TEXT<IY,6>') '
602     RETADD = IDENT(TEXT<IY,5>,'') IY + 1
603     IY = IY + 1 :((EE4)
604 EE7   AYREF = 0
605     OPND = TEXT<IY,4>
606     IDENT(OPND) :S(EE9)
607     OPND SPAN(DIGIT) RPOS(0) :S(EE7D)
608     OPND BREAK('+-') . OPND1 LEN(1) . X0 SPAN(DIGIT) . OPND2 RPOS(0)
609     :S(EE7A)
610     IX = SYMBOL<OPND>
611     IDENT(IX) :S($E 'EE1')
612     SYMTBL<IX,1> ('V' | 'A' | 'C') . X0 :F($E 'EE1')
613     OPND1 = IDENT(X0,'A') OPND :F(EE7D)
614     OPND2 = '0' :((EE8)
615 EE7A  IX = SYMBOL<OPND1>
616     IDENT(IX) :S($E 'EE7A')
617     SYMTBL<IX,1> 'A' :S(EE8)
618     SYMTBL<IX,1> ('L' | 'V') . X1 :F($E 'EE7A1')
619     IW = IDENT(X1,'L') SYMTBL<IX,3>
620     IW = IDENT(X1,'V') SYMTBL<IX,2>
621     IW = EVAL(IW ' ' X0 ' ' OPND2) :F($E 'EE7A2')
622     I = 1
623 EE7B  EQ(SYMTBL<I,2> ,IW) :S(EE7C)

```

```

622      I = LT(I,SYMX) I + 1 :F($ (E *EE7B*))S(EE7B)
623 EE7C SYMTBL<I,1> 'V' :F($ (E *EE7B*))
624      OPND = SYMTBL<I,3> :S(EE9)
625 EE7D IDENT(TEXT<IY,3>) :S(EE9)
626      REST = INST(TEXT<IY,3>,OPND)
627      STMT = 'X' REST 'X'
628      OUTSTMT(STMT)
629      CPTH = CPTH :; STMT
630      REST = SUB(REST)
631      REST 'DIR($AQ' . X0 BREAK(') . X1 *' = X0 (X1 + 1) *'
632      CON = DIFFER(CON) CON ' REST :S(EE9)
633      CON = REST :S(EE9)
634 EE8      TEMP = SYMTBL<SYMBOL<OPND1>,1>
635      IQ = 0
636      IDENT(TEMP) :S($ (E *EE8*))
637      TEMP 'A;' = :F($ (E *EE9*))
638 EE8A      REST = '0.LE.' OPND2 'AND.' OPND2 'LT.' TEMP
639      STMT = 'X' REST 'X'
640      OUTSTMT(STMT)
641      CPTH = CPTH :; STMT
642      CON = DIFFER(CON) ' SUB(REST) :S(EE8C)
643      CON = SUB(REST)
644 EE8C      OPND2 = NE(IQ,0) OPND2 - 1 :S(EE8B)
645      TEXT<IY,2> ('SAQ' | 'LAQ' ) :F(EE8B)
646      IQ = 1
647      OPND2 = OPND2 + 1 :S(EE8A)
648 EE8B AYREF = 1
649      AYNAM = OPND1
650      AYDIM = TEMP - 1
651      AYDISP = OPND2
652 EE9      TEMP = TEXT<IY,5>
653      IDENT(TEXT<IY,2>) :S(EE9A)
654      STMT = TEXT<IY,2>
655      OUTSTMT(STMT)
656      CPTH = CPTH :; STMT
657      X1 = SYMBOL<TEXT<IY,4>>
658      SS = CONVERT(SYMTBL<X1,5>,'INTEGER')
659      SAC = CONVERT(SAC,'INTEGER')
660      SMQ = CONVERT(SMQ,'INTEGER')
661      TEXT<IY,2> BREAK(') . X0
662      X0 ('ADD' | 'SUB') . OP RPOS(0) :F(EE90)
663      OP = '+'
664      OP = IDENT(X0,'SUB') '-'
665      ST =
666      EQ(SAC,SS) :F($ (E *EE90*))
667 EE9C      ACTYPE = DIFFER(SYMTBL<X1,6>,'ACTYPE') 'A'
668      IDENT(ACTYPE,'A') :S(EE90B)
669      REST = 'R($AC' OP OPND '))
670      STMT = 'X' REST 'X'
671      OUTSTMT(STMT)
672      CPTH = CPTH :; STMT
673      REST = SUB(REST)
674      CON = DIFFER(CON) CON ' SUB(REST) :S(EE90A)
675      CON = SUB(REST)
676 EE90A      TEMP = 'SAC=$AC' OP OPND :S(EE9A)
677 EE90B      TEMP = 'SAC=N($AC' OP OPND ') ST :S(EE9A)

```

```

678 EE90 X0 ('ADS' | 'SBS') . OP RPOS(0) :F(EE91)
679 OP = '+'
680 OP = IDENT(X0,'SBS') '-'
681 EQ(SAC,SS) :F($E 'EE90'))
682 ST = ',' OPND '=' SAC : (EE9C)
683 EE91 IDENT(X0,'MPY') :F(EE92)
684 ACTYPE = IDENT(SYMTBL<X1,6> ,ACTYPE) IDENT(ACTYPE,'I') 'I'
+ :F(EE91A)
685 MQTYPE = ACTYPE : (EE91B)
686 EE91A ACTYPE = 'A'
687 MQTYPE = 'A'
688 EE91B SAC = SAC + SS
689 SMQ = SAC - 23 : (EE9A)
690 EE92 IDENT(X0,'DIV') :F(EE93)
691 IDENT(SYMTBL<X1,6> , 'A') :S($E 'EE92'))
692 SAC = SAC - SS : (EE9A)
693 EE93 IDENT(X0,'LDA') :F(EE94)
694 ACTYPE = SYMTBL<X1,6>
695 SAC = SS : (EE9A)
696 EE94 IDENT(X0,'LDQ') :F(EE95)
697 MQTYPE = SYMTBL<X1,6>
698 SMQ = SS : (EE9A)
699 EE95 IDENT(X0,'STA') :F(EE96)
700 IDENT(SYMTBL<X1,6> ,ACTYPE) :F($E 'EE95'))
701 EQ(SS,SAC) :F($E 'EE95'))S(EE9A)
702 EE96 IDENT(X0,'STQ') :F(EE97)
703 IDENT(SYMTBL<X1,6> ,MQTYPE) :F($E 'EE96'))
704 EQ(SS,SMQ) :F($E 'EE96'))S(EE9A)
705 EE97 X0 POS(0) ('ANA' | 'ORA') RPOS(0) :F(EE98)
706 EQ(SS,SAC) :F($E 'EE97'))S(EE9A)
707 EE98 IDENT('XAQ',X0) :F(EE99)
708 IX = SAC
709 SAC = SMQ
710 SMQ = IX : (EE9A)
711 EE99 X2 = CONVERT(TEXT<IY,4> , 'INTEGER') :F(EE9A)
712 SAC = IDENT(X0,'SAR') SAC + X2 :S(EE9A)
713 SAC = IDENT(X0,'SLR') SAC + X2 :S(EE9A)
714 SAC = IDENT(X0,'SAL') SAC - X2 :S(EE9A)
715 SAC = IDENT(X0,'SLL') SAC - X2 :S(EE9A)
716 EE9A TEMP DIFFER(TEMP) BAL . X ('.' | RPOS(0)) = :F(EE4B)
717 X BREAK(' ') . X0 '=' REM . X1 :F($E 'EE9A'))
718 X0 '(' BAL . X0A '.' BAL . X0B ')' RPOS(0) :S(EE10)
719 EQ(AYREF,1) :S(EE9D)
720 EE9B X1 = INST(X1,OPND)
721 X1 = SUB(X1)
722 X0 = INST(X0,OPND)
723 IX = SYMBOL<X0>
724 SYMTBL<IX,4> = SYMTBL<IX,4> + 1
725 X0 = SUB(X0)
726 X2 = X0 '=' X1
727 HYP = HYP '.' X2 : (EE9A)
728 EE9D X1 = INST(X1,AYNAM '(' AYDISP ')')
729 X1 = SUB(X1)
730 IDENT(X0,'#') :S(EE9E)
731 X0 = SUB(X0)
732 X2 = X0 '=' X1

```

```

733      HYP = HYP * X2 : (EE9A)
734 EE9E  X0 = INST(X0,AYNAM *(' AYDISP '))
735      IX = SYMBOL<AYNAM>
736      ODIS = SYMTBL<IX,4>
737      SYMTBL<IX,4> = ODIS + 1
738      X0 = SUB(X0)
739      X2 = *FI(Z,0,* AYDIM *,(Z,NE,* AYDISP *,'IMP,' SUB(AYNAM)
      * (Z)=* AYNAM ODIS *(Z).AND,* X0 *=' X1 *))
740      HYP = HYP * X2 : (EE9A)
741 EE10  X1 *(' BAL * X1A *,' BAL * X1B *) RPOS(0) :F($E 'EE10'))
742      IDENT(X0A,##') IDENT(X0B,##') :S(EE11)
743      IDENT(X1A,##') IDENT(X1B,##') :S(EE12)
744      EQ(AYREF,1) :S(EE10B)
745      X3 = SUB(INST(X1B,OPND))
746      X2 = SUB(INST(X1A,OPND))
747 EE10A IX = SYMBOL<X0B>
748      SYMTBL<IX,4> = SYMTBL<IX,4> + 1
749      X1 = SUB(X0B)
750      IX = SYMBOL<X0A>
751      SYMTBL<IX,4> = SYMTBL<IX,4> + 1
752      X0 = SUB(X0A)
753      HYP = HYP * X0 *=' X2 *,' X1 *=' X3 : (EE9A)
754 EE10B X3 = SUB(INST(X1B,AYNAM *(' AYDISP ')) )
755      X2 = SUB(INST(X1A,AYNAM *(' AYDISP ')) ) : (EE10A)
756 EE11  OUTPUT = 'ARRAY REFERENCES HAS NOT BEEN IMPLEMENTED' : (END)
      *****
      * ERROR MESSAGES
      *****
757
758 EREVLIX1 OUTPUT = 'EREVLIX1' : (END)
759 EREVLIX2 OUTPUT = 'EREVLIX2' : (END)
760 ERCV3A OUTPUT = 'ERCV3A' : (END)
761 ERDD27A OUTPUT = 'ERDD27A' : (END)
762 ERDD27B OUTPUT = 'ERDD27B' : (END)
763 ERDD27 OUTPUT = 'ERDD27' : (END)
764 ERCONV1 OUTPUT = 'ERCONV1' : (END)
765 ERBB3 OUTPUT = 'ERBB3' : (END)
766 ERBB2 OUTPUT = 'ERBB2' : (END)
767 ERBB1 OUTPUT = 'ERBB1' : (END)
768 ERBC3A OUTPUT = 'ERBC3A' : (END)
769 ERBSS0 OUTPUT = 'ERBSS0' : (END)
770 ERBSS1 OUTPUT = 'ERBSS1' : (END)
771 ERBSS2 OUTPUT = 'ERBSS2' : (END)
772 ERBEN0 OUTPUT = 'ERBEN0' : (END)
773 ERGEN1 OUTPUT = 'ERGEN1' : (END)
774 ERGEN2 OUTPUT = 'ERGEN2' : (END)
775 EREQU1 OUTPUT = 'EREQU1' : (END)
776 EREQU2 OUTPUT = 'EREQU2' : (END)
777 ERBC1 OUTPUT = 'ERBC1' : (END)
778 ERN01 OUTPUT = 'ERN01' : (END)
779 ERBC2 OUTPUT = 'ERBC2' : (END)
780 ERBC3 OUTPUT = 'ERBC3' : (END)
781 ERBC5 OUTPUT = 'ERBC5' : (END)
782 ERBC6 OUTPUT = 'ERBC6' : (END)
783 EREE7A1 OUTPUT = 'EREE7A1' : (END)
784 EREE0 OUTPUT = 'EREE0' : (END)

```

```
785 EREE1 OUTPUT = 'EREE1'  :(END)
786 EREE7A OUTPUT = 'EREE7A'  :(END)
787 EREE7A2 OUTPUT = 'EREE7A2'  :(END)
788 EREE11B OUTPUT = 'EREE11B'  :(END)
789 ERMP1 OUTPUT = 'ERMP1'  :(END)
790 ERMP2 OUTPUT = 'ERMP2'  :(END)
791 ERMP3 OUTPUT = 'ERMP3'  :(END)
792 EREE7B OUTPUT = 'EREE7B'  :(END)
793 EREE9 OUTPUT = 'EREE9'  :(END)
794 EREE9A OUTPUT = 'EREE9A'  :(END)
795 ERCB3B OUTPUT = 'ERCV3B'  :(END)
796 ERVC3C OUTPUT = 'ERCV3C'  :(END)
797 ERVC4A OUTPUT = 'ERCV4A'  :(END)
798 ERVC4B OUTPUT = 'ERCV4B'  :(END)
799 ERVC4C OUTPUT = 'ERCV4C'  :(END)
800 ERSE OUTPUT = 'ERSE'  :(END)
801 EREE90 OUTPUT = 'EREE90'  :(END)
802 EREE95 OUTPUT = 'EREE95'  :(END)
803 EREE97 OUTPUT = 'EREE97'  :(END)
804 EREE99 OUTPUT = 'EREE99'  :(END)
805 EREE92 OUTPUT = 'EREE92'  :(END)
806 ERDD13 OUTPUT = 'ERDD13'  :(END)
807 ERDD14 OUTPUT = 'ERDD14'  :(END)
808 ERDD21 OUTPUT = 'ERDD11'  :(END)
809 END
```

--- SPECIFICATION ---

A0=0,A1200=1200,B23,C180RF=180,B15,C382=3.82,CSH15=5.B15.
 DEG2=0.01111111B0,DEG4=0.02222222B0,DEG6=0.03333333B0,DEG15=0.08333333B0.
 DEG20=0.11111111B0,DEG25=0.13888888B0,DELTAT=0.05H0,DR,DELTAT=0.1B0.
 FDBKG=0.8B3,ABS(GAMMA).LT.1.B0,ABS(GAMMAC).LT.0.00555555B0,ABS(GAMCMD).LT.1.B0.
 ABS(GS).LE.0.625B15,ABS(GSA).LT.1.B0,ABS(GSDEV).LE.0.00111111B6.
 HRA0.GE.0.AND.HRA0.LT.1500.B23,HREF.GE.50.B15.AND.HREF.LT.1.B0.
 ABS(HDCF).LE.1.B8.
 KDAMP=0.9B0,ABS(KHD).LT.1.B0.AND.A3S(KHD).GT.0.016B0,ABS(KFLR).LE.1.B15.
 K1=700,B11,K2=3.B15,K3=0.03B0,K4=0,K5=0.39B0,K6=0.653B0,K7=2.B15.
 K8=-1.6H5,K10=-3.B15,K1000=837.B23,K1200=926.B23.
 ABS(LATSTR).LT.1.B0,LGAMC.GE.-0.00555555B0.AND.LGAMC.LE.0.
 LIMY=400.B23,LIMLOC=100.B23,LIMPR=0.00333333B0,LIM180=003777777.
 ABS(LOCDEV).LE.0.01111111B6,ABS(LOCSTR).LT.1.B0,MDA=250.B23.
 ABS(PCMD).LE.0.03333333B0,PERIOD=35.B10,PID2=3.141592682.
 ABS(PITCH).LT.1.B0,ABS(PSIF).LT.1.B0,ABS(RCMD).LE.0.13888888B0.
 ABS(ROLL).LT.1.B0,ABS(RWYHDG).LT.1.B0,RESET=000000000,SET=011111111.
 TAU1=3.B3,TAU2=60.B6,TAU3=1.B2,TAU4=3.84,TAU5=1.B3,TAU6=1.5B3.
 TF=16.B5,TIMER.GE.0.AND.TIMER.LT.1.B0,ABS(TK).LT.1.B0,ABS(TP1).LE.1.B3.
 ABS(TP2).LE.1.B3,ABS(TP3).LE.1.B13,ABS(TP4).LT.1.B0,ABS(TP5).LT.1.B0.
 ABS(TP6).LT.1.B0,ABS(TP7).LE.1.B3,T12=0400,A3S(VERSTR).LE.1.B8,VGS.GE.100.B15.
 ABS(YETA).LT.1.B0,ABS(RCMD).LE.0.13888888B0,BINTME.GE.0.AND.BINTME.LT.1.B0.
 MSKL15=000077777,ABS(RLFD).LT.1.B0

ORIGINAL PAGE IS
 OF POOR QUALITY

--- SCALING ---

{SET,C,23,0},{T12,C,8,0},{TMEFLG,V,B,0},{PCUMWD,V,B,0},{MPILS,V,B,0}
 {LSON,V,B,0},{GRUND,V,B,0},{START,V,B,0},{A1200,C,23,1}
 {LOCDF,V,B,0},{LOCOC,V,B,0},{FLARE,V,B,0},{K1,C,11,1}
 {GPLOCD,V,12,1},{YETA,V,23,1},{TK,V,0,A},{RWYHOG,V,0,A},{PSIE,V,0,A}
 {DEG2,C,0,A},{LIMLOC,C,23,1},{GUID20,V,B,0},{LBS,V,B,0},{ROLL,V,0,A}
 {RLFD,V,0,A},{LIMY,C,23,1},{PERIOD,C,10,1},{K5,C,0,1},{TP1,V,3,A},{TAU1,C,3,1}
 {DELTAT,V,0,1},{GS,V,15,1},{K6,C,0,1},{KDAMP,C,0,1},{TP2,V,3,A},{HRAD,V,23,1}
 {K1000,C,23,1},{TP7,V,3,A},{TAU2,C,6,1},{LIM180,C,3,A},{LOCSTR,V,0,A}
 {DEG20,C,0,A},{CAPT,V,B,0},{A0,C,0,A},{K1200,C,23,1},{RCMVLD,V,B,0}
 {PCMVLD,V,B,0},{LATSTR,V,0,A},{DEG25,C,0,A},{RCMD,V,0,A},{TAU3,C,2,1}
 {LOCVLD,V,B,0},{DEG4,C,0,A},{GSFD,V,B,0},{GSBSNS,V,B,0},{GAMMA,C,0,A}
 {LGAMC,V,0,A},{GSDEV,V,12,A},{TP3,V,15,A},{TAU4,C,4,1},{C180BF,C,15,1}
 {K3,C,0,1},{GSA,V,0,A},{K4,C,0,1},{TP4,V,0,A},{HDCF,V,15,1},{K2,C,15,1}
 {K10,C,15,1},{TF,C,5,1},{K7,C,15,1},{K8,C,5,1},{HREF,V,15,1},{KFLR,V,8,1}
 {MSKL15,C,8,1},{LGAMC,V,0,A},{TAU5,C,3,1},{LIMPR,C,0,A},{GAMCMD,V,0,A}
 {PITCH,V,0,A},{TP6,V,0,A},{TAU6,C,3,1},{TPS,V,0,A},{GAMMAC,V,0,A},{VGS,V,15,1}
 {KHO,V,0,1},{PIB2,C,2,1},{C3B2,C,2,1},{MDA,C,23,1},{TIMER,V,23,1}
 {BINTME,V,23,1},{FDBKG,C,3,1},{DEG5,C,0,A},{PCMD,V,0,A},{GSVLD,V,B,0}
 {VERSTR,V,15,1},{C5B15,C,15,1},{LOCDEV,V,12,1},{DEG15,C,0,A},{GUID30,V,B,0}

--- LISTING OF THE PROGRAM TO BE VERIFIED ---

```

REM *****
REM  FLIGHT DIRECTOR ROUTINE
REM *****
REM
* CEXIT0.SPEC
EADIFD SEX  EFDRTN
LDA  TMEFLG      REAL TIME ?
TRZ  EXIT1
LDA  PCUMWD      EXTRACT BIT FOR
ANA  T12          ILS SELECTED ON
STA  MPILS        EADI MODE PANEL
LDA  MPILS        ***** USE LAND MODE WHEN AVAILABLE *****
TRZ  **5
LDA  ILSON        ILS TUNED ?
TRZ  **3
LDA  GRUND        NOT GO AROUND ?
TRZ  **5
STZ  START
STZ  LOCFD
STZ  GSFD
TRA  CRUZFD
LDA  START        FIRST PASS ?
TNZ  FD1
LDA  SET          START=MPILS X ILSON X (-GRUND)
STA  START        INITIALIZE FOR
STZ  TP1          FOR FIRST PASS
STZ  TP2
STZ  TP3
STZ  TP4
STZ  TP5
STZ  TP6
STZ  TP7
STZ  LOCFD
STZ  LOCOC
STZ  FLARE

* CEXIT0.SPEC.
* CFD1,MPILS=MPILS01,TP1=TP102,TP2=TP202,TP3=TP302,TP4=TP402,TP5=TP502,
* TP6=TP602,TP7=TP702,LOCFD=LOCFD03,LOCOC=LOCOC02,FLARE=FLARE02,START=START03
* ,GSFD=GSFD02,,NOT.CFDIN1,TMEFLG.NE.0
FD1  LDA  LOCDEV
MPY  K1
STA  YETA
LDA  TK
SUB  RWHOG        COMPUTE PSI
STA  PSIE         ERROR
ABS
SUB  DEG2         WITHIN LIMITS ?
TRM  FD3
LDA  YETA
ABS
SUB  LIMLOC       LAT DEV WITHIN LIMITS ?
TRM  FD3
LDA  GUID2D       2D GUIDANCE ON ?
TRZ  FO4
FD2  LDA  LOCFD    BRANCH LOC/CRUISE F.D.
TNZ  LANDED
TRA  CRUZFD
FD3  LDA  SET

```

* CFD4,MPILS=MPILS01,TP1=TP102,TP2=TP202,TP3=TP302,TP4=TP402,TP5=TP502,
 * TP6=TP602,TP7=TP702,LOCFD=LOCFD03,LOCCC=LOCC04,FLARE=FLARE02,START=START03,
 * YETA=YETA01,PSIE=PSIE01,GSFD=GSFD02,.NOT.CFDINI,TMEFLG.NE.0

FD4 LDA LBS LOC BEAM SENSED ?
 TRZ FD2
 LDA LOCFD TEST FOR FIRST PASS
 TNZ LANDFD
 LDA SET INITIALIZE
 STA LOCFD RLFD
 LDA ROLL AS FD BARS
 STA RLFD COME UP

* CEXIT0,SPEC.
 * CLANDFD,MPILS=MPILS01,TP1=TP102,TP2=TP202,TP3=TP302,TP4=TP402,TP5=TP502,
 * TP6=TP602,TP7=TP702,LOCFD=LOCFD05,LOCCC=LOCC04,FLARE=FLARE02,START=START03,
 * RLFD=RLFD02,YETA=YETA01,PSIE=PSIE01,GSFD=GSFD02,.NOT.CFDINI,TMEFLG.NE.0

* ..NOT.CCRUZF
 LANDFD LDA LOCCC LOCALIZER ON COURSE
 TRZ FD5 NO. SKIP COURSE CUT LIMITER
 LDA YETA
 LDD LIMY
 TRA SRLMT
 TRA **2

FD5 LDA YETA YETA OR LIMITED YETA
 DIV PERIOD
 DIV PERIOD
 MPY K5 COMPUTATION OF
 SUB TP1
 DIV TAU1 $TP1 = (YETA \times K5 / PER**2) / ((TAU1 \times S<61)$
 MPY DELTAT
 SAR 3
 ADS TP1
 LDA GS *****
 DIV PERIOD
 MPY K6 COMPUTATION OF
 MPY PSIE
 MPY KDAMP $(GS \times PSIE \times KDAMP \times K6)/PERIOD$
 SLL 2
 STA TP2

* CEXIT0,SPEC.
 * CLANDFD,MPILS=MPILS01,TP1=TP103,TP2=TP203,TP3=TP302,TP4=TP402,TP5=TP502,
 * TP6=TP602,TP7=TP702,LOCFD=LOCFD05,LOCCC=LOCC04,FLARE=FLARE02,START=START03,
 * RLFD=RLFD02,YETA=YETA01,PSIE=PSIE01,GSFD=GSFD02,.NOT.CFDINI,TMEFLG.NE.0

* ..NOT.CCRUZF
 LDA HRAD *****
 SUB K1000
 TRP FD6
 LDA TP2
 SUB TP7 WASHOUT IF
 STA TP2 HRAD LESS
 MPY DELTAT THAN 1000 FT
 DIV TAU2
 SAR 6
 ADS TP7

* CEXIT0,SPEC.
 * CFD6,MPILS=MPILS01,TP1=TP103,TP2=TP205,TP3=TP302,TP4=TP402,TP5=TP502,
 * TP6=TP602,TP7=TP704,LOCFD=LOCFD05,LOCCC=LOCC04,FLARE=FLARE02,START=START03,
 * RLFD=RLFD02,YETA=YETA01,PSIE=PSIE01,GSFD=GSFD02,.NOT.CFDINI,TMEFLG.NE.0

* ..NOT.CCRUZF
 FD6 LDA TP2 *****
 SUB TP1
 LDD LIM180
 TRA SRLMT
 SAL 3
 CMA
 STA LOCTR

```

MPY PSIE LE ZERO ?
TRZ **2
TRP FD7
LDA LOCSTR LOCSTR X PSIE
MPY PSIE GE ZERO ?
TRM FD7
LDA PSIE PSIE GT 20
ABS
SUB DEG20
TRM FD7
STZ CAPT
LDA A0
TRA FD11
* CEXIT0.SPEC.
* CFD7.MPILS=MPILS01.TP1=TP103.TP2=TP205.TP3=TP302.TP4=TP402.TP5=TP502.
* TP6=TP602.TP7=TP704.LOCFD=LOCFD05.LOCOC=LOCOC04.FLARE=FLARE02.START=START03.
* RLFD=RLFD02.YETA=YETA01.PSIE=PSIE01.GSFD=GSFD02..NOT.CFDIN1.TMEFLG.NE.0
* .LOCSTR=LOCSTR01..NOT.CCRUZF0
FD7 LDA SET
STA CAPT
LDA HRAD HRAD GT 1200?
SUB A1200
TRM FD8 LESS THAN 1200
LDA LOCSTR
TRA FD10
FD8 LDA LOCSTR
LDQ DEG15
TRA SRLMT LIMIT LOCSTR AT LOW ALT.
TRA FD11
* CEXIT0.SPEC.
* CCRUZF0.START=START04.LOCFD=LOCFD06.GSFD=GSFD02.MPILS=MPILS01.TP1=TP102.
* TP2=TP202.TP3=TP302.TP4=TP402.TP5=TP502.TP6=TP602.TP7=TP702.
* LOCOC=LOCOC04.FLARE=FLARE02.YETA=YETA02.PSIE=PSIE02.TMEFLG.NE.0
* ...NOT.CFD42
CRUZF0 LDA GUID2D CRUISE F.D.
TNZ FD9 GUID2D ON ?
EXIT1 STZ RCMVLD BOTH ROLL AND
STZ PCMVLD PITCH COMMANDS INVALID
* .FALSE.
TRA* EFDRTN
FD9 LDA LATSTR
FD10 LDQ DEG25
TRA SRLMT
FD11 SUB RLFD
SAR 2
DIV TAUJ
LDQ DEG4 LIMIT ROLL RATE
TRA SRLMT
MPY DELTAT
ADS RLFD
SUB ROLL
LDQ DEG25
TRA SRLMT
STA RCM0
* CEXIT0.SPEC.
* CFD11.START=START04.LOCFD=LOCFD06.GSFD=GSFD02.MPILS=MPILS01.TP1=TP104.
* TP2=TP206.TP3=TP302.TP4=TP402.TP5=TP502.TP6=TP602.TP7=TP705.FLARE=FLARE02.
* LOCOC=LOCOC04.YETA=YETA02.PSIE=PSIE02.LOCSTR=LOCSTR02.CAPT=CAPT04.
* RLFD=RLFD04.RCMD=RCMD01
LDA LOCFD *****
TRZ FD13
LDA LOCVLD ROLL COMMAND
TRZ FD14
LDA MPILS VALIDITY TEST
TRZ FD14

```

COMMAND VALID

```

STA RCMVLD
TRA VERTFD
FD13 LDA GUID2D
TNZ FD12
FD14 STZ RCMVLD *****
* CEXIT0, SPEC.
* CFD11, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104,
* TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02,
* LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04,
* RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04
VERTFD LDA LOCFD TEST FOR ILS FLT DIR
TRZ VCRUZ GO TO CRUISE MODE
VLAND LDA GSFD TEST FOR FIRST PASS
TNZ FD15 NOT FIRST PASS
LDA GSBSNS TEST FOR GLIDE SLOPE BEAM SENSED
TRZ VCRUZ GLIDE SLOPE NOT SENSED GO TO CRUISE
LDA SET FIRST PASS IN GLIDE SLOPE
STA GSFD SET GSFD HI AND
LDA GAMMA INITIALIZE LGAMC TO ROLL
STA LGAMC WHEN FD BARS COME UP
FD15 LDA GSDEV *****GPGSDV=GSDEV***
MPY K1 COMPUTE GAMMA CMD
SLL 8
SUB TP3
DIV TAU4  $TP4 = (GSDEV \times K1 / (TAU4 \times 1)) \times K3 / 180 + GSA - K4$ 
MPY DELTAT
SAR 4
ADS TP3
DIV C180BF
MPY K3
ADD GSA GLIDE SLOPE ANGLE
SUB K4
CMA
STA TP4 *****
* CEXIT0, SPEC.
* CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104,
* TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02,
* LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04,
* RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02
LDA FLARE FLARE LATCHED ?
TNZ FLRLT *****
LDA HD CF LIMIT HD CF
ADD K2
TRM **3 TO LESS
LDA K10 THAN -3
TRA **2
LDA HD CF
ADD K7 COMPUTE
MPY TF HREF
DIV K8 IF FLARE
STA HREF NOT LATCHED
* CEXIT0, SPEC.
* CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104,
* TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02,
* LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04,
* RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02,
* HREF=HREF02
FLRLT LDA HRAD COMPARE HRAD AND HREF
DIV HREF FOR FLARE LATCHING
STA KFLR AND FLARE GAIN COMPUTATIONS
SUB MSKL15 HRAD/HREF
TRP **4 LESS THAN ONE
LDA SET YES, SET
STA FLARE FLARE LATCH
TRA **2

```

```

* CEXIT0,SPEC.
* CFD15,START=START04,LOCFD=LOCFD06,MPILS=MPILS01,TP1=TP104,
* TP2=TP206,TP3=TP303,TP4=TP403,TP5=TP502,TP6=TP602,TP7=TP705,FLARE=FLARE03,
* LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
* RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04,GSFD=GSFD03,LGAMC=LGAMC02,
* HREF=HREF02,KFLR=KFLR01

```

```

LDA KFLR
LDQ MSKL15
TRA SRLMT
SAL 8
MPY TP4
SUB K4
TRM **3
LDA A0
TRA **9
STA GAMMAC
ACD GSA
ADD GSA
TRM **3
LDA GAMMAC
TRA **3
SUB GAMMAC
CMA
SUB LGAMC
DIV TAU5
SAR 3
LDQ LIMPR
TRA SRLMT
MPY DELTAT
ADS LGAMC
SUB GAMMA
STA GAMCMD

```

```

*****
LIMIT GAMMA COMMAND
BETWEEN ZERO AND MINUS 2*GSA

```

```

*****
INTEGRATE GAMMA COMMAND
LIMIT PITCH RATE

```

```

* CEXIT0,SPEC.
* CFD15,START=START04,LOCFD=LOCFD06,MPILS=MPILS01,TP1=TP104,
* TP2=TP206,TP3=TP303,TP4=TP403,TP5=TP502,TP6=TP602,TP7=TP705,FLARE=FLARE03,
* LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
* RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04,GSFD=GSFD03,GAMCMD=GAMCMD01,
* HREF=HREF02,KFLR=KFLR01,GAMMAC=GAMMAC01,LGAMC=LGAMC03

```

```

LDA PITCH
SUB TP5
STA TP6
MPY DELTAT
DIV TAU6
SAR 3
ADS TP5
LOA TP6
MPY FDBKG
SLL 3
CMA
ADD GAMCMD
LDQ DEG6
TRA SRLMT
STA PCMD

```

```

*****
COMPUTE
WASHOUT
TIME
CONSTANT
*****
COMPUTE PITCH COMMAND
LIMIT PITCH CMD TO 6 DEG

```

```

* CEXIT0,SPEC.
* CFD15,START=START04,LOCFD=LOCFD06,MPILS=MPILS01,TP1=TP104,
* TP2=TP206,TP3=TP303,TP4=TP403,TP5=TP503,TP6=TP603,TP7=TP705,FLARE=FLARE03,
* LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
* RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04,GSFD=GSFD03,LGAMC=LGAMC03,
* HREF=HREF02,GAMMAC=GAMMAC01,PCMD=PCMD01,KFLR=KFLR01,GAMCMD=GAMCMD01

```

```

LDA GSVLD
TRZ EXIT2
EXIT3 LDA RCMVL
STA PCMVL
* FALSE.
TRA* FORTN

```

```

GS TUNED ?
PITCH COMMAND IS VALID IF
ROLL COMMAND IS VALID

```

```

TRZ  EXIT2      NO. BLANK VERT COMMAND
LDA  VERSTR     FETCH 3D GUIDANCE STEERING SIGNAL
LDQ  C5H15      5 FPS**2 LIMIT
TRA  SRLMT
DIV  VGS
SAR  2
DIV  KHD
DIV  PIB2
LDQ  DEG6      LIMIT PCMD TO 6 DEG
TRA  SRLMT
SUB  PCMD
DIV  C302
MPY  DELTAT
SAR  2
ADS  PCMD
LDA  MDA      *****
SUB  HRAD
TRA  EXIT3      MDA TEST
LDA  TIMER
SUB  BINTME     FLASH FLT. DIR.
* CEXIT0.PCMD=PCMD02.RCMVLD=RCMVLD04.RCMD=RCMD01.RLFD=RLFD04.
* START=START04.LOCFO=LOCFO06.GSFD=GSFD02.MPILS=MPILS01.TP1=TP104.
* TP2=TP206.TP3=TP302.TP4=TP402.TP5=TP502.TP6=TP602.TP7=TP705.FLARE=FLARE02.
* LOCCC=LOCC04.YETA=YETA02.PSIE=PSIE02.LOCSTR=LOCSTR02.CAPT=CAPT04
* .KFLR=KFLR01
  TRZ* EFDRTN      BARS IF
  LDA  BINTME      BELOW MINIMUM
  STA  TIMER      DESCENT ALTITUDE
  LDA  PCMVLD
  TRZ  EXIT3      *****
EXIT2 STZ  PCMVLD  PITCH COMMAND INVALID
* .FALSE.
  TRA* EFDRTN

```


ABBREVIATION : SPEC IS

A0=0.A1200=1200.H23.C180HF=180.B15.C3B2=3.B2.C5B15=5.B15.DEG2=0.01111111B0.DEG4=0.02222222B0.DEG6=0.03333333B0.DEG15=0.09333333B0.DE
G20=0.11111111B0.DEG25=0.13888888B0.DELTAT=0.05B0.DR.DELTAT=0.180.FDBKG=0.8B3.ABS(GAMMA).LT.1.B0.ABS(GAMMAC).LT.0.00555555B0.ABS(GAM
CMD).LT.1.B0.ABS(GS).LE.0.625B15.ABS(GSA).LT.1.B0.ABS(GSDEV).LE.0.00111111B6.HRAD.GE.0.AND.HRAD.LT.1500.B23.HREF.GE.50.B15.AND.HREF.
LT.1.B0.ABS(HDCF).LE.1.B9.KDAMP=0.990.ABS(KHD).LT.1.B0.AND.ABS(KHD).GT.0.016B0.ABS(KFLR).LE.1.B15.K1=700.B11.K2=3.315.K3=0.03B0.K4=0
.K5=0.39B0.K6=0.65330.K7=2.815.K8=-1.6B5.K10=-3.B15.K1000=837.B23.K1200=925.B23.ABS(LATSTR).LT.1.B0.LGAMC.GE.-0.00555555B0.AND.LGAMC
.LE.0.LIMY=400.B23.LINLOC=100.B23.LIMPR=0.00333333B0.LIM180=00377777.ABS(LOCDEV).LE.0.01111111B6.ABS(LOCSTR).LT.1.B0.MDA=250.B23.AB
S(PCMD).LE.0.03333333B0.PERIOD=35.H10.P1B2=3.1415926B2.ABS(PITCH).LT.1.B0.ABS(PSIE).LT.1.B0.ABS(RCMD).LE.0.13888888B0.ABS(ROLL).LT.1
.B0.ABS(RWYHDG).LT.1.B0.RESET=000000000.SET=011111111.TAU1=3.83.TAU2=60.B6.TAU3=1.B2.TAU4=3.B4.TAU5=1.B3.TAU6=1.5B3.TF=15.35.TIMER.G
E.0.AND.TIMER.LT.1.B0.ABS(TK).LT.1.30.ABS(TP1).LE.1.B3.ABS(TP2).LE.1.B3.ABS(TP3).LE.1.B13.ABS(TP4).LT.1.B0.ABS(TP5).LT.1.30.ABS(TP6)
.LT.1.B0.ABS(TP7).LE.1.B3.T12=0400.ABS(VERSTR).LE.1.B8.VGS.GE.100.B15.ABS(YETA).LT.1.B0.ABS(RCMD).LE.0.13888888B0.BINTME.GE.0.AND.BI
NTME.LT.1.B0.MSKL15=000077777.ABS(RLFD).LT.1.B0

ORIGINAL PAGE IS
OF POOR QUALITY

--- CONTROL PATH 1 ---

```
*CEXIT0, SPEC*
SEX EFDRTN      LDA TMEFLG      .NOT.($AC.EQ.0)    LDA PCUMWD      ANA T12      STA MPILS
LDA MPILS      .NOT.($AC.EQ.0)    LDA ILSON      .NOT.($AC.EQ.0)    LDA GRUND      .NOT.($AC.EQ.0)
STZ START      STZ LOCFD      STZ GSFD
*CEXIT0, SPEC, CCRUZFD, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG.NE.0, .NOT.CFD42*
```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	.NOT.(\$AC1.EQ.0)	\$AC2=PCUMWD
\$AC3=AND(\$AC2,T12)	MPILS1=\$AC3	\$AC4=MPILS1	.NOT.(\$AC4.EQ.0)	\$AC5=ILSON	.NOT.(\$AC5.EQ.0)
\$AC6=GRUND	.NOT.(\$AC6.EQ.0)	START1=0	LOCFD1=0	GSFD1=0	
.IMPLIES.					
CCRUF0	START1=START04	LOCFD1=LOCFD06	GSFD1=GSFD02	MPILS1=MPILS01	TP1=TP102
TP2=TP202	TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702
LOCOC=LOCOC04	FLARE=FLARE02	YETA=YETA02	PSIE=PSIE02	TMEFLG.NE.0	.NOT.CFD42

--- CONTROL PATH 2 ---

```
*CEXIT0, SPEC*
SEX EFDRTN      LDA TMEFLG      .NOT.($AC.EQ.0)    LDA PCUMWD      ANA T12      STA MPILS
LDA MPILS      .NOT.($AC.EQ.0)    LDA ILSON      .NOT.($AC.EQ.0)    LDA GRUND      ($AC.EQ.0)
LDA START      .NOT.($AC.NE.0)    LDA SET      STA START      STZ TP1
STZ TP3      STZ TP4      STZ TP5      STZ TP6      STZ TP7
STZ LOCOC      STZ FLARE
*CEXIT0, SPEC, CF01, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFD=LOCFD03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0*
```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	.NOT.(\$AC1.EQ.0)	\$AC2=PCUMWD
\$AC3=AND(\$AC2,T12)	MPILS1=\$AC3	\$AC4=MPILS1	.NOT.(\$AC4.EQ.0)	\$AC5=ILSON	.NOT.(\$AC5.EQ.0)
\$AC6=GRUND	\$AC6.EQ.0	\$AC7=START	.NOT.(\$AC7.NE.0)	\$AC8=SET	START1=\$AC8
TP11=0	TP21=0	TP31=0	TP41=0	TP51=0	TP61=0
TP71=0	LOCFD1=0	LOCOC1=0	FLARE1=0		
.IMPLIES.					
CF01	MPILS1=MPILS01	TP11=TP102	TP21=TP202	TP31=TP302	TP41=TP402
TP51=TP502	TP61=TP602	TP71=TP702	LOCFD1=LOCFD03	LOCOC1=LOCOC02	FLARE1=FLARE02
START1=START03	GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0		

--- CONTROL PATH 3 ---

```
*CEXIT0, SPEC*
SEX EFDRTN      LDA TMEFLG      .NOT.($AC.EQ.0)    LDA PCUMWD      ANA T12      STA MPILS
LDA MPILS      .NOT.($AC.EQ.0)    LDA ILSON      .NOT.($AC.EQ.0)    LDA GRUND      ($AC.EQ.0)
LDA START      ($AC.NE.0)
*CEXIT0, SPEC, CF01, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFD=LOCFD03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0*
```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	.NOT.(\$AC1.EQ.0)	\$AC2=PCUMWD
\$AC3=AND(\$AC2,T12)	MPILS1=\$AC3	\$AC4=MPILS1	.NOT.(\$AC4.EQ.0)	\$AC5=ILSON	.NOT.(\$AC5.EQ.0)
\$AC6=GRUND	\$AC6.EQ.0	\$AC7=START	\$AC7.NE.0		
.IMPLIES.					
CF01	MPILS1=MPILS01	TP1=TP102	TP2=TP202	TP3=TP302	TP4=TP402
TP5=TP502	TP6=TP602	TP7=TP702	LOCFD=LOCFD03	LOCOC=LOCOC02	FLARE=FLARE02
START=START03	GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0		

```

*CEXIT0, SPEC*
SEX EFDRTN LDA TMEFLG .NOT. ($AC.EQ.0) LDA PCUMWD ANA T12 STA MPILS
LDA MPILS .NOT. ($AC.EQ.0) LDA ILSON ($AC.EQ.0) STZ START STZ LOCFD
STZ GSFD
*CEXIT0, SPEC, CCRUZFD, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG.NE.0, .NOT.CFD42*

```

*** VERIFICATION CONDITIONS ***

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	.NOT. (\$AC1.EQ.0)	\$AC2=PCUMWD
\$AC3=AND (\$AC2, T12)	MPILS1=\$AC3	\$AC4=MPILS1	.NOT. (\$AC4.EQ.0)	\$AC5=ILSON	\$AC5.EQ.0
START1=0	LOCFD1=0	GSFD1=0			
.IMPLIES.					
CCRUFZFD	START1=START04	LOCFD1=LOCFD06	GSFD1=GSFD02	MPILS1=MPILS01	TP1=TP102
TP2=TP202	TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702
LOCOC=LOCOC04	FLARE=FLARE02	YETA=YETA02	PSIE=PSIE02	TMEFLG.NE.0	.NOT.CFD42

--- CONTROL PATH 5 ---

```

*CEXIT0, SPEC*
SEX EFDRTN LDA TMEFLG .NOT. ($AC.EQ.0) LDA PCUMWD ANA T12 STA MPILS
LDA MPILS ($AC.EQ.0) STZ START STZ LOCFD STZ GSFD
*CEXIT0, SPEC, CCRUFZFD, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG.NE.0, .NOT.CFD42*

```

*** VERIFICATION CONDITIONS ***

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	.NOT. (\$AC1.EQ.0)	\$AC2=PCUMWD
\$AC3=AND (\$AC2, T12)	MPILS1=\$AC3	\$AC4=MPILS1	\$AC4.EQ.0	START1=0	LOCFD1=0
GSFD1=0					
.IMPLIES.					
CCRUFZFD	START1=START04	LOCFD1=LOCFD06	GSFD1=GSFD02	MPILS1=MPILS01	TP1=TP102
TP2=TP202	TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702
LOCOC=LOCOC04	FLARE=FLARE02	YETA=YETA02	PSIE=PSIE02	TMEFLG.NE.0	.NOT.CFD42

--- CONTROL PATH 6 ---

```

*CEXIT0, SPEC*
SEX EFDRTN LDA TMEFLG ($AC.EQ.0) STZ RCMVLD STZ PCMVLD
*.FALSE.*

```

*** VERIFICATION CONDITIONS ***

CEXIT0	SPEC	EFDRTN1=\$RA	\$AC1=TMEFLG	\$AC1.EQ.0	RCMVLD1=0
PCMVLD1=0					
.IMPLIES.					
.FALSE.					

--- CONTROL PATH 7 ---

```

*CEXIT0, SPEC, CFD1, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFD=LOCFD03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, .NOT.CFDINI, TMEFLG.NE.0*
LDA LOCDEV MPY K1 STA YETA LDA TK SUB RYHGD STA PSIE
ABS SUB DEG2 .NOT. ($AC.LT.0) LDA YETA ABS SUB LIMLOC SUB LIMLOC
XR ($AC-LIMLOC)X .NOT. ($AC.LT.0) LDA GUID2D .NOT. ($AC.EQ.0) LDA LOCFD .NOT. ($AC.NE.0)
*CEXIT0, SPEC, CCRUFZFD, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG.NE.0, .NOT.CFD42*

```

*** VERIFICATION CONDITIONS ***

CEXIT0	SPEC	CFD1	MPILS=MPILS01	TP1=TP102	TP2=TP202
--------	------	------	---------------	-----------	-----------

TP3=TP302 LOCOC=LOCOC02 \$AC1=LOCDEV PSIE1=\$AC4 \$AC9=\$AC8-LIMLOC *IMPLIES. R(\$AC8-LIMLOC) PSIE1=PSIE02	TP4=TP402 FLARE=FLARE02 \$M01=MULTM(\$AC1,K1) \$AC5=ABS(\$AC4) *NOT.(\$AC9.LT.0) CCRUZFD *NOT.CFD42	TP5=TP502 START=START03 \$AC2=MULT(\$AC1,K1) \$AC6=N(\$AC5-DEG2) \$AC10=GUID20 START=START04	TP6=TP602 GSFD=GSFD02 YETA1=\$AC2 *NOT.(\$AC6.LT.0) *NOT.(\$AC10.EQ.0) LOCDF=LOCDF06	TP7=TP702 *NOT.CFDIN1 \$AC3=TK \$AC7=YETA1 \$AC11=LOCDF LOCOC=LOCOC04	LOCDF=LJCFD03 TMEFLG.NE.0 \$AC4=N(\$AC3-RWYHDG) \$AC8=ABS(\$AC7) *NOT.(\$AC11.NE.0) YETA1=YETA02
---	---	---	---	--	---

--- CONTROL PATH 8 ---

*CEXIT0, SPEC, CFD1, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0*

LDA LOCDEV	MPY K1	STA YETA	LDA TK	SUB RWYHDG	STA PSIE
ABS	SUB DEG2	*NOT.(\$AC.LT.0)	LDA YETA	ABS	SUB LIMLOC
XR(\$AC-LIMLOC)%	*NOT.(\$AC.LT.0)	LDA GUID20	*NOT.(\$AC.EQ.0)	LDA LOCDF	(\$AC.NE.0)

*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LJCFD05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0, *NOT.CCRUZFD*

** VERIFICATION CONDITIONS **

CEXIT0 TP3=TP302 LOCOC=LOCOC02 \$AC1=LOCDEV PSIE1=\$AC4 \$AC9=\$AC8-LIMLOC *IMPLIES. R(\$AC8-LIMLOC) PSIE1=PSIE01	SPEC TP4=TP402 FLARE=FLARE02 \$M01=MULTM(\$AC1,K1) \$AC5=ABS(\$AC4) *NOT.(\$AC9.LT.0) CLANDFD *NOT.CCRUZFD	CFD1 TP5=TP502 START=START03 \$AC2=MULT(\$AC1,K1) \$AC6=N(\$AC5-DEG2) \$AC10=GUID20 LOCDF=LOCDF05	MPILS=MPILS01 TP6=TP602 GSFD=GSFD02 YETA1=\$AC2 *NOT.(\$AC6.LT.0) *NOT.(\$AC10.EQ.0) LOCOC=LOCOC04	TP1=TP102 TP7=TP702 *NOT.CFDIN1 \$AC3=TK \$AC7=YETA1 \$AC11=LOCDF RLFD=RLFD02	TP2=TP202 LOCDF=LJCFD03 TMEFLG.NE.0 \$AC4=N(\$AC3-RWYHDG) \$AC8=ABS(\$AC7) \$AC11.NE.0 YETA1=YETA01
---	---	---	--	---	---

--- CONTROL PATH 9 ---

*CEXIT0, SPEC, CFD1, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0*

LDA LOCDEV	MPY K1	STA YETA	LDA TK	SUB RWYHDG	STA PSIE
ABS	SUB DEG2	*NOT.(\$AC.LT.0)	LDA YETA	ABS	SUB LIMLOC
XR(\$AC-LIMLOC)%	*NOT.(\$AC.LT.0)	LDA GUID20	(\$AC.EQ.0)		

*CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0*

** VERIFICATION CONDITIONS **

CEXIT0 TP3=TP302 LOCOC=LOCOC02 \$AC1=LOCDEV PSIE1=\$AC4 \$AC9=\$AC8-LIMLOC *IMPLIES. R(\$AC8-LIMLOC)	SPEC TP4=TP402 FLARE=FLARE02 \$M01=MULTM(\$AC1,K1) \$AC5=ABS(\$AC4) *NOT.(\$AC9.LT.0) CFD4	CFD1 TP5=TP502 START=START03 \$AC2=MULT(\$AC1,K1) \$AC6=N(\$AC5-DEG2) \$AC10=GUID20 LOCOC=LOCOC04	MPILS=MPILS01 TP6=TP602 GSFD=GSFD02 YETA1=\$AC2 *NOT.(\$AC6.LT.0) \$AC10.EQ.0 YETA1=YETA01	TP1=TP102 TP7=TP702 *NOT.CFDIN1 \$AC3=TK \$AC7=YETA1 PSIE1=PSIE01	TP2=TP202 LOCDF=LJCFD03 TMEFLG.NE.0 \$AC4=N(\$AC3-RWYHDG) \$AC8=ABS(\$AC7)
---	--	---	--	--	--

--- CONTROL PATH 10 ---

*CEXIT0, SPEC, CFD1, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0*

LDA LOCDEV	MPY K1	STA YETA	LDA TK	SUB RWYHDG	STA PSIE
ABS	SUB DEG2	*NOT.(\$AC.LT.0)	LDA YETA	ABS	SUB LIMLOC
XR(\$AC-LIMLOC)%	(\$AC.LT.0)	LDA SET	STA LOCOC		

*CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, *NOT.CFDIN1, TMEFLG.NE.0*

** VERIFICATION CONDITIONS **

ORIGINAL PAGE IS
OF POOR QUALITY

CEXIT0
TP3=TP302
LOCOC=LOCOC02
\$AC1=LOCDEV
PSIE1=\$AC4
\$AC9=\$AC8-LIMLOC
.IMPLIES.
R(\$AC8-LIMLOC)

SPEC
TP4=TP402
FLARE=FLARE02
\$MQ1=MULTM(\$AC1,K1)
\$AC5=ABS(\$AC4)
\$AC9.LT.0

CFD4

CFD1
TP5=TP502
START=START03
\$AC2=MULT(\$AC1,K1)
\$AC6=N(\$AC5-DEG2)
\$AC10=SET

LOCOC1=LOCOC04

MPILS=MPILS01
TP6=TP602
GSFD=GSFD02
YETA1=\$AC2
.NOT.(\$AC6.LT.0)
LOCOC1=\$AC10

YETA1=YETA01

TP1=TP102
TP7=TP702
.NOT.CFDIN1
\$AC3=TK
\$AC7=YETA1

PSIE1=PSIE01

TP2=TP202
LOCFC=LOCFC03
TMEFLG.NE.0
\$AC4=N(\$AC3-RWYHDG)
\$AC8=ABS(\$AC7)

--- CONTROL PATH 11 ---

CEXIT0, SPEC, CFD1, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC03, LOCOC=LOCOC02, FLARE=FLARE02, START=START03, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0
LDA LOCDEV MPY K1 STA YETA LDA TK SUB RWYHDG STA PSIE
ABS SUB DEG2 (\$AC.LT.0) LDA SET STA LOCOC
CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0

** VERIFICATION CONDITIONS **

CEXIT0
TP3=TP302
LOCOC=LOCOC02
\$AC1=LOCDEV
PSIE1=\$AC4

SPEC
TP4=TP402
FLARE=FLARE02
\$MQ1=MULTM(\$AC1,K1)
\$AC5=ABS(\$AC4)

LOCOC1=LOCOC04

CFD1
TP5=TP502
START=START03
\$AC2=MULT(\$AC1,K1)
\$AC6=N(\$AC5-DEG2)

YETA1=YETA01

MPILS=MPILS01
TP6=TP602
GSFD=GSFD02
YETA1=\$AC2
\$AC6.LT.0

PSIE1=PSIE01

TP1=TP102
TP7=TP702
.NOT.CFDIN1
\$AC3=TK
\$AC7=SET

TP2=TP202
LOCFC=LOCFC03
TMEFLG.NE.0
\$AC4=N(\$AC3-RWYHDG)
LOCOC1=\$AC7

.IMPLIES.
CFD4

--- CONTROL PATH 12 ---

CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0
LDA LBS .NOT.(\$AC.EQ.0) LDA LOCFC .NOT.(\$AC.NE.0) LDA SET STA LOCFC
LDA ROLL STA RLFD
CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUFZD

** VERIFICATION CONDITIONS **

CEXIT0
TP3=TP302
LOCOC=LOCOC04
.NOT.CFDIN1
\$AC3=SET
.IMPLIES.
CLANDFD

SPEC
TP4=TP402
FLARE=FLARE02
TMEFLG.NE.0
LOCFC1=\$AC3

LOCFC1=LOCFC05

CFD4
TP5=TP502
START=START03
\$AC1=LBS
\$AC4=ROLL

RLFD1=RLFD02

MPILS=MPILS01
TP6=TP602
YETA=YETA01
.NOT.(\$AC1.EQ.0)
RLFD1=\$AC4

.NOT.CCRUFZD

TP1=TP102
TP7=TP702
PSIE=PSIE01
\$AC2=LOCFC

TP2=TP202
LOCFC=LOCFC03
GSFD=GSFD02
.NOT.(\$AC2.NE.0)

--- CONTROL PATH 13 ---

CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0
LDA LBS .NOT.(\$AC.EQ.0) LDA LOCFC (\$AC.NE.0)
CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUFZD

** VERIFICATION CONDITIONS **

CEXIT0
TP3=TP302
LOCOC=LOCOC04
.NOT.CFDIN1

SPEC
TP4=TP402
FLARE=FLARE02
TMEFLG.NE.0

LOCFC=LOCFC03

CFD4
TP5=TP502
START=START03
\$AC1=LBS

MPILS=MPILS01
TP6=TP602
YETA=YETA01
.NOT.(\$AC1.EQ.0)

LOCFC=LOCFC03

TP1=TP102
TP7=TP702
PSIE=PSIE01
\$AC2=LOCFC

TP2=TP202
LOCFC=LOCFC03
GSFD=GSFD02
\$AC2.NE.0

.IMPLIES.
CLANDFD

--- CONTROL PATH 14 ---

CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0
 LDA LBS (\$AC.EQ.0) LDA LOCDF .NOT.(\$AC.NE.0)
 CEXIT0, SPEC, CCRUF0, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG.NE.0, .NOT.CFD4

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD4	MPILS=MPILS01	TP1=TP102	TP2=TP202
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF03
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	YETA=YETA01	PSIE=PSIE01	GSFD=GSFD02
.NOT.CFDIN1	TMEFLG.NE.0	SAC1=LBS	SAC1.EQ.0	SAC2=LOCDF	.NOT.(\$AC2.NE.0)
.IMPLIES.					
CCRUF0	START=START04	LOCDF=LOCDF06	YETA=YETA02	PSIE=PSIE02	.NOT.CFD4*

--- CONTROL PATH 15 ---

CEXIT0, SPEC, CFD4, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF03, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0
 L/JA LBS (\$AC.EQ.0) LDA LOCDF (\$AC.NE.0)
 CEXIT0, SPEC, CLANDF0, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUF0

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD4	MPILS=MPILS01	TP1=TP102	TP2=TP202
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF03
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	YETA=YETA01	PSIE=PSIE01	GSFD=GSFD02
.NOT.CFDIN1	TMEFLG.NE.0	SAC1=LBS	SAC1.EQ.0	SAC2=LOCDF	SAC2.NE.0
.IMPLIES.					
CLANDF0	LOCDF=LOCDF05	RLFD=RLFD02	.NOT.CCRUF0		

--- CONTROL PATH 16 ---

CEXIT0, SPEC, CLANDF0, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUF0
 LDA LOCOC .NOT.(\$AC.EQ.0) LDA YETA LDQ LIMY TRA SRLMT
 %ABS(\$AC).LT.ABS(PERIOD)% DIV PERIOD %ABS(\$AC).LT.ABS(PERIOD)% DIV PERIOD
 MPY K5 SUB TP1 %ABS(\$AC).LT.ABS(TAU1)% DIV TAU1
 SAR 3 ADS TP1 LDA GS %ABS(\$AC).LT.ABS(PERIOD)% STA TP2
 MPY K6 MPY PSIE MPY KDAMP XDIR(\$AQ*2**2)* SLL 2
 CEXIT0, SPEC, CLANDF0, MPILS=MPILS01, TP1=TP103, TP2=TP203, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUF0

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CLANDF0	MPILS=MPILS01	TP1=TP102	TP2=TP202
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUF0	SAC1=LOCOC	.NOT.(\$AC1.EQ.0)
SAC2=YETA	\$MQ1=LIMY	\$AC3=SRLMT(\$AC2,\$MQ1)	\$AC4=\$AC3/PERIOD	SAC5=\$AC4/PERIOD	\$MQ2=MULTM(\$AC5,K5)
\$AC6=MULT(\$AC5,K5)	\$AC7=N(\$AC5-TP1)	\$AC8=\$AC7/TAU1	\$MQ3=MULTM(\$AC8,DELTA)	\$AC11=N(\$AC10+TP1)	\$AC12=GS
\$AC9=MULT(\$AC8,DELTA)	\$MQ4=MULTM(\$AC13,K6)	\$AC10=\$AC9/2**3	\$AC11=N(\$AC10+TP1)	TP11=\$AC11	
\$AC13=\$AC12/PERIOD	\$MQ5=MULTM(\$AC14,PSIE)	\$AC14=MULT(\$AC13,K6)	\$MQ5=MULTM(\$AC14,PSIE)	\$AC16=MULT(\$AC15,KDAMP)	
\$AC15=MULT(\$AC14,PSIE)	\$MQ6=MULTM(\$AC15,KDAMP)	\$MQ6=MULTM(\$AC15,KDAMP)	\$MQ6=MULTM(\$AC15,KDAMP)	TP21=\$AC17	
\$AQ1=(\$AC16,\$MQ6)	\$AQ2=\$AQ1*2**2	\$AC17=AR(\$AQ2)	\$MQ7=MR(\$AQ2)		
.IMPLIES.					
ABS(\$AC3).LT.ABS(PERIOD)		ABS(\$AC4).LT.ABS(PERIOD)		ABS(\$AC7).LT.ABS(TAU1)	
ABS(\$AC12).LT.ABS(PERIOD)		DIR(\$AQ1*2**2)	TP11=TP103	TP21=TP203	

```

*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04
, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*
LDA LOCOC ($AC.E0.0) LDA YETA XABS($AC).LT.ABS(PERIOD)% DIV PERIOD
XABS($AC).LT.ABS(PERIOD)% DIV PERIOD MPY K5 SUB TP1
XABS($AC).LT.ABS(TAU1)% DIV TAU1 MPY DELTAT SAR 3 ADS TP1
LDA GS XABS($AC).LT.ABS(PERIOD)% DIV PERIOD MPY K6 MPY PSIE
MPY KDAMP XDIR($AQ*2**2)% SLL 2 STA TP2
*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP103, TP2=TP203, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04
, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CLANDFD	MPILS=MPILS01	TP1=TP102	TP2=TP202
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=LOCOC	SAC1.EQ.0
SAC2=YETA	SAC3=\$AC2/PERIOD	SAC4=\$AC3/PERIOD	\$M01=MULTM(\$AC4,K5)	SAC5=MULT(\$AC4,K5)	SAC6=N(\$AC5-TP1)
SAC7=\$AC6/TAU1	\$M02=MULTM(\$AC7,DELTAT)	SAC11=GS	SAC8=MULT(\$AC7,DELTAT)	\$M03=MULTM(\$AC12,K6)	SAC9=\$AC8/2**3
SAC10=N(\$AC9+TP1)	TP11=\$AC10	SAC14=MULT(\$AC13,PSIE)	SAC12=\$AC11/PERIOD	\$M05=MULTM(\$AC14,KDAMP)	SAC13=MULT(\$AC12,K6)
\$M04=MULTM(\$AC13,PSIE)	SAC15=MULT(\$AC14,KDAMP)	\$AQ1=(\$AC15,\$M05)	\$AQ2=\$AQ1*2**2	SAC16=AR(\$AQ2)	\$M06=MR(\$AQ2)
TP21=\$AC16	.IMPLIES.	ABS(\$AC3).LT.ABS(PERIOD)	ABS(\$AC6).LT.ABS(TAU1)		
ABS(\$AC2).LT.ABS(PERIOD)	ABS(\$AC11).LT.ABS(PERIOD)	DIR(\$AQ1*2**2)	TP11=TP103	TP21=TP203	

--- CONTROL PATH 18 ---

```

*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP103, TP2=TP203, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04
, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*
LDA HRAD SUB K1000 X($AC-K1000)% .NOT.($AC.GE.0) LDA TP2 SUB TP7
STA TP2 MPY DELTAT XABS($AC).LT.ABS(TAU2)% DIV TAU2 SAR 6
ADS TP7
*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CLANDFD	MPILS=MPILS01	TP1=TP103	TP2=TP203
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=HRAD	SAC2=\$AC1-K1000
.NOT.(\$AC2.GE.0)	SAC3=TP2	SAC4=N(\$AC3-TP7)	TP21=\$AC4	\$M01=MULTM(\$AC4,DELTAT)	
SAC5=MULT(\$AC4,DELTAT)	SAC6=\$AC5/TAU2	SAC7=\$AC6/2**6	SAC8=N(\$AC7+TP7)	TP71=\$AC8	
.IMPLIES.					
R(\$AC1-K1000)	ABS(\$AC5).LT.ABS(TAU2)	CFD6	TP21=TP205	TP71=TP704	

--- CONTROL PATH 19 ---

```

*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP103, TP2=TP203, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCDF=LOCDF05, LOCOC=LOCOC04
, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*
LDA HRAD SUB K1000 X($AC-K1000)% ($AC.GE.0)
*CEXIT0, SPEC, CLANDFD, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CLANDFD	MPILS=MPILS01	TP1=TP103	TP2=TP203
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP702	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=HRAD	SAC2=\$AC1-K1000
SAC2.GE.0					

R(\$AC1-K1000)

CFD6

TP2=TP205

TP7=TP704

--- CONTROL PATH 20 ---

```

*CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT. CFDIN1, TMEFLG.NE.0, .NOT. CCRUFZD*
LDA TP2 SUB TP1 LDQ LIM180 TRA SRLMT %R($AC2**3)% SAL 3
CMA STA LOCSTR LDA YETA MPY PSIE .NOT.($AC.EQ.0) .NOT.($AC.GE.0)
LDA LOCSTR MPY PSIE .NOT.($AC.LT.0) LDA PSIE ABS SAR 2 SUB DEG20
.NOT.($AC.LT.0) STZ CAPT LDA A0 SUB RLFD TRA SRLMT MPY DELTAT
XABS($AC).LT.ABS(TAU3)% DIV TAU3 LDQ DEGA STA RCMD
ADS RLFD SUB ROLL LDQ DEG25 TRA SRLMT
*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602
, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT. CFDIN1	TMEFLG.NE.0	.NOT. CCRUFZD	SAC1=TP2	SAC2=N(\$AC1-TP1)
\$M01=LIM180	SAC3=SRLMT(\$AC2,\$M01)	SAC4=\$AC3*2**3	SAC5=-\$AC4	LOCSTR1=\$AC5	SAC6=YETA
\$M02=MULTM(\$AC6,PSIE)	SAC7=MULT(\$AC6,PSIE)	.NOT.(\$AC7.EQ.0)	.NOT.(\$AC7.GE.0)	SAC8=LOCSTR1	\$M03=MULTM(\$AC8,PSIE)
SAC9=MULT(\$AC8,PSIE)	.NOT.(\$AC9.LT.0)	SAC10=PSIE	SAC11=ABS(\$AC10)	SAC12=N(\$AC11-DEG20)	.NOT.(\$AC12.LT.0)
CAPT1=0	SAC13=A0	SAC14=N(\$AC13-RLFD)	SAC15=\$AC14/2**2	SAC16=\$AC15/TAU3	\$M04=DEG4
\$AC17=SRLMT(\$AC16,\$M04)		\$M05=MULTM(\$AC17,DELTAT)		SAC18=MULT(\$AC17,DELTAT)	
\$AC19=N(\$AC18+RLFD)	RLFD1=\$AC19	\$AC20=N(\$AC19-ROLL)	\$M06=DEG25	SAC21=SRLMT(\$AC20,\$M06)	
RCMD1=\$AC21					
.IMPLIES.					
R(\$AC3*2**3)	ABS(\$AC15).LT.ABS(TAU3)		CFD11	START=START04	LOCDF=LOCDF06
TP1=TP104	TP2=TP206	TP7=TP705	YETA=YETA02	PSIE=PSIE02	LOCSTR1=LOCSTR02
CAPT1=CAPT04	RLFD1=RLFD04	RCMD1=RCMD01			

--- CONTROL PATH 21 ---

```

*CEXIT0, SPEC, CFD5, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT. CFDIN1, TMEFLG.NE.0, .NOT. CCRUFZD*
LDA TP2 SUB TP1 LDQ LIM180 TRA SRLMT %R($AC2**3)% SAL 3
CMA STA LOCSTR LDA YETA MPY PSIE .NOT.($AC.EQ.0) .NOT.($AC.GE.0)
LDA LOCSTR MPY PSIE .NOT.($AC.LT.0) LDA PSIE ABS SUB DEG20
($AC.LT.0)
*CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT. CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT. CCRUFZD*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT. CFDIN1	TMEFLG.NE.0	.NOT. CCRUFZD	SAC1=TP2	SAC2=N(\$AC1-TP1)
\$M01=LIM180	SAC3=SRLMT(\$AC2,\$M01)	SAC4=\$AC3*2**3	SAC5=-\$AC4	LOCSTR1=\$AC5	SAC6=YETA
\$M02=MULTM(\$AC6,PSIE)	SAC7=MULT(\$AC6,PSIE)	.NOT.(\$AC7.EQ.0)	.NOT.(\$AC7.GE.0)	SAC8=LOCSTR1	\$M03=MULTM(\$AC8,PSIE)
SAC9=MULT(\$AC8,PSIE)	.NOT.(\$AC9.LT.0)	SAC10=PSIE	SAC11=ABS(\$AC10)	SAC12=N(\$AC11-DEG20)	SAC12.LT.0
.IMPLIES.					
R(\$AC3*2**3)	CFD7	LOCSTR1=LOCSTR01			

--- CONTROL PATH 22 ---

```

*CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT. CFDIN1, TMEFLG.NE.0, .NOT. CCRUFZD*
LDA TP2 SUB TP1 LDQ LIM180 TRA SRLMT %R($AC2**3)% SAL 3
CMA STA LOCSTR LDA YETA MPY PSIE .NOT.($AC.EQ.0) .NOT.($AC.GE.0)
LDA LOCSTR MPY PSIE ($AC.LT.0)
*CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT. CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT. CCRUFZD*

```

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCFC=LOCFC05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=TP2	SAC2=N(\$AC1-TP1)
\$M01=LIM180	\$AC3=SRLMT(\$AC2,\$M01)	\$AC4=\$AC3*2**3	\$AC5=-\$AC4	LOCSTR1=\$AC5	\$AC6=YETA
\$M02=MULTM(\$AC6,PSIE)	\$AC7=MULT(\$AC6,PSIE)	.NOT.(\$AC7.EQ.0)	.NOT.(\$AC7.GE.0)	SAC9=LOCSTR1	\$M03=MULTM(\$AC8,PSIE)
\$AC9=MULT(\$AC8,PSIE)	\$AC9.LT.0				
.IMPLIES.					
R(\$AC3*2**3)	CFD7	LOCSTR1=LOCSTR01			

--- CONTROL PATH 23 ---

CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD

LDA TP2	SUB TP1	LDO LIM180	TRA SRLMT	XR(\$AC2**3)X	SAL 3
CMA	STA LOCSTR	LDA YETA	MPY PSIE	.NOT.(\$AC.EQ.0)	(\$AC.GE.0)

CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUZFD

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCFC=LOCFC05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=TP2	SAC2=N(\$AC1-TP1)
\$M01=LIM180	\$AC3=SRLMT(\$AC2,\$M01)	\$AC4=\$AC3*2**3	\$AC5=-\$AC4	LOCSTR1=\$AC5	\$AC6=YETA
\$M02=MULTM(\$AC6,PSIE)	\$AC7=MULT(\$AC6,PSIE)	.NOT.(\$AC7.EQ.0)	\$AC7.GE.0		
.IMPLIES.					
R(\$AC3*2**3)	CFD7	LOCSTR1=LOCSTR01			

--- CONTROL PATH 24 ---

CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUZFD

LDA TP2	SUB TP1	LDO LIM180	TRA SRLMT	XR(\$AC2**3)X	SAL 3
CMA	STA LOCSTR	LDA YETA	MPY PSIE	(\$AC.EQ.0)	LDA LOCSTR
MPY PSIE	.NOT.(\$AC.LT.0)	LDA PSIE	ABS	SUB DEG20	.NOT.(\$AC.LT.0)
ST7 CAPT	LDA A0	SUB RLFD	SAR 2	%ABS(\$AC).LT.ABS(TAU3)X	
D1 TAU3	LDO DEG4	TRA SRLMT	MPY DELTAT	ADS RLFD	SUB ROLL
LDO DEG25	TRA SRLMT	STA RCMD			

CEXIT0, SPEC, CFD11, START=START04, LOCFC=LOCFC05, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCFC=LOCFC05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUZFD	SAC1=TP2	SAC2=N(\$AC1-TP1)
\$M01=LIM180	\$AC3=SRLMT(\$AC2,\$M01)	\$AC4=\$AC3*2**3	\$AC5=-\$AC4	LOCSTR1=\$AC5	\$AC6=YETA
\$M02=MULTM(\$AC6,PSIE)	\$AC7=MULT(\$AC6,PSIE)	\$AC7.EQ.0	\$AC8=LOCSTR1	\$M03=MULTM(\$AC8,PSIE)	\$AC9=MULT(\$AC8,PSIE)
.NOT.(\$AC9.LT.0)	\$AC10=PSIE	\$AC11=ARS(\$AC10)	\$AC12=N(\$AC11-DEG20)	.NOT.(\$AC12.LT.0)	CAPT1=0
\$AC13=A0	\$AC14=N(\$AC13-RLFD)	\$AC15=\$AC14/2**2	\$AC16=\$AC15/TAU3	\$M04=DEG4	
\$AC17=SRLMT(\$AC16,\$M04)	RLFD1=\$AC19	\$M05=MULTM(\$AC17,DELTAT)	\$AC21=SRLMT(\$AC20,\$M06)	\$M18=MULT(\$AC17,DELTAT)	
\$AC19=N(\$AC18*RLFD)		\$AC20=N(\$AC19-ROLL)	\$M06=DEG25		
RCMD1=\$AC21					
.IMPLIES.					
R(\$AC3*2**3)	ABS(\$AC15).LT.ABS(TAU3)		CFD11	START=START04	LOCFC=LOCFC05
TP1=TP104	TP2=TP206	TP7=TP705	YETA=YETA02	PSIE=PSIE02	LOCSTR1=LOCSTR02
CAPT1=CAPT04	RLFD1=RLFD04	RCMD1=RCMD01			

--- CONTROL PATH 25 ---

```

*CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUFZD*
LDA TP2 SUB TP1 LDQ LIM180 TRA SRLMT XR(SAC*2**3)% SAL 3
CMA STA LOCSTR LDA YETA MPY PSIE (SAC.EQ.0) LDA LOCSTR
MPY PSIE .NOT.(SAC.LT.0) LDA PSIE AUS SUB DEG20 (SAC.LT.0)
*CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*

```

**** VERIFICATION CONDITIONS ****

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUFZD	SAC1=TP2	SAC2=N(SAC1-TP1)
\$M01=LIM180	SAC3=SRLMT(SAC2,\$M01)	SAC4=SAC3*2**3	SAC5=-SAC4	LOCSTR1=SAC5	SAC6=YETA
\$M02=MULTM(SAC6,PSIE)	SAC7=MULT(SAC6,PSIE)	SAC7.EQ.0	SAC8=LOCSTR1	\$M03=MULTM(SAC8,PSIE)	SAC9=MULT(SAC8,PSIE)
.NOT.(SAC9.LT.0)	SAC10=PSIE	SAC11=ABS(SAC10)	SAC12=N(SAC11-DEG20)	SAC12.LT.0	
.IMPLIES.					
R(SAC3*2**3)	CFD7	LOCSTR1=LOCSTR01			

--- CONTROL PATH 26 ---

```

*CEXIT0, SPEC, CFD6, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUFZD*
LDA TP2 SUB TP1 LDQ LIM180 TRA SRLMT XR(SAC*2**3)% SAL 3
CMA STA LOCSTR LDA YETA MPY PSIE (SAC.EQ.0) LDA LOCSTR
MPY PSIE (SAC.LT.0)
*CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*

```

**** VERIFICATION CONDITIONS ****

CEXIT0	SPEC	CFD6	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	.NOT.CCRUFZD	SAC1=TP2	SAC2=N(SAC1-TP1)
\$M01=LIM180	SAC3=SRLMT(SAC2,\$M01)	SAC4=SAC3*2**3	SAC5=-SAC4	LOCSTR1=SAC5	SAC6=YETA
\$M02=MULTM(SAC6,PSIE)	SAC7=MULT(SAC6,PSIE)	SAC7.EQ.0	SAC8=LOCSTR1	\$M03=MULTM(SAC8,PSIE)	SAC9=MULT(SAC8,PSIE)
SAC9.LT.0					
.IMPLIES.					
R(SAC3*2**3)	CFD7	LOCSTR1=LOCSTR01			

--- CONTROL PATH 27 ---

```

*CEXIT0, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*
LDA SET STA CAPT LDA HRAD SUB A1200 XR(SAC-A1200)% .NOT.(SAC.LT.0)
LDA LOCSTR LDQ DEG25 TRA SRLMT SUB RLFD SAR 2
XA05(SAC).LT.ABS(TAU3)% DIV TAU3 LDQ DEG4 TRA SRLMT MPY DELTAT
ADS RLFD SUB ROLL LDQ DEG25 TRA SRLMT STA RCM0
*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602
TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCM0=RCM001*

```

**** VERIFICATION CONDITIONS ****

CEXIT0	SPEC	CFD7	MPILS=MPILS01	TP1=TP103	TP2=TP205
TP3=TP302	TP4=TP402	TP5=TP502	TP6=TP602	TP7=TP704	LOCDF=LOCDF05
LOCOC=LOCOC04	FLARE=FLARE02	START=START03	RLFD=RLFD02	YETA=YETA01	PSIE=PSIE01
GSFD=GSFD02	.NOT.CFDIN1	TMEFLG.NE.0	LOCSTR=LOCSTR01	.NOT.CCRUFZD	SAC1=SET
CAPT1=SAC1	SAC2=HRAD	SAC3=SAC2-A1200	.NOT.(SAC3.LT.0)	SAC4=LOCSTR	\$M01=DEG25
SAC5=SRLMT(SAC4,\$M01)	SAC6=N(SAC5-RLFD)	SAC7=SAC6/2**2	SAC8=SAC7/TAU3	\$M02=DEG4	SAC9=SRLMT(SAC9,\$M02)
\$M03=MULTM(SAC9,DELTAT)		SAC10=MULT(SAC9,DELTAT)		SAC11=N(SAC10+RLFD)	RLFD1=SAC11
SAC12=N(SAC11-ROLL)	\$M04=DEG25	SAC13=SRLMT(SAC12,\$M04)		RCM01=SAC13	
.IMPLIES.					
R(SAC2-A1200)	A13(SAC7).LT.ABS(TAU3)				

ORIGINAL PAGE IS
OF POOR QUALITY

```

LCA TP2      SUB TP1      LDQ LIM180      TRA MPY      %R($AC*2**3)X      SAL 3
CMA          STA LOCSTR   LDA YETA      ABS          ($AC.EQ.0)      LDA LOCSTR
MPY PSIE     .NOT.($AC.LT.0)      LDA PSIE      SUB DEG20      ($AC.LT.0)
*CEXIT0, SPEC, CF07, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*

```

```

** VERIFICATION CONDITIONS **

CEXIT0      SPEC      CF06      MPILS=MPILS01      TP1=TP103      TP2=TP205
TP3=TP302    TP4=TP402    TP5=TP502    TP6=TP602    TP7=TP704    LOCDF=LOCDF05
LOCOC=LOCOC04    FLARE=FLARE02    START=START03    RLFD=RLFD02    YETA=YETA01    PSIE=PSIE01
GSFD=GSFD02      .NOT.CFDIN1    TMEFLG.NE.0    .NOT.CCRUFZD    $AC1=TP2      $AC2=N($AC1-TP1)
$M01=LIM180      $AC3=SRLMT($AC2,$M01)    $AC4=$AC3*2**3    $AC5=-$AC4    LOCSTR1=$AC5    $AC6=YETA
$M02=MULTM($AC6,PSIE)    $AC7=MULT($AC6,PSIE)    $AC7.EQ.0      $AC8=LOCSTR1    $M03=MULTM($AC8,PSIE)    $AC9=MULT($AC8,PSIE)
.NOT.($AC9.LT.0)    $AC10=PSIE      $AC11=ABS($AC10)    $AC12=N($AC11-DEG20)    $AC12.LT.0
.IMPLIES.
R($AC3*2**3)      CF07      LOCSTR1=LOCSTR01

```

```

--- CONTROL PATH 26 ---

*CEXIT0, SPEC, CF06, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, .NOT.CCRUFZD*

LDA TP2      SUB TP1      LDQ LIM180      TRA SRLMT      %R($AC*2**3)X      SAL 3
CMA          STA LOCSTR   LDA YETA      MPY PSIE          ($AC.EQ.0)      LDA LOCSTR
MPY PSIE     ($AC.LT.0)
*CEXIT0, SPEC, CF07, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*

```

```

** VERIFICATION CONDITIONS **

CEXIT0      SPEC      CF06      MPILS=MPILS01      TP1=TP103      TP2=TP205
TP3=TP302    TP4=TP402    TP5=TP502    TP6=TP602    TP7=TP704    LOCDF=LOCDF05
LOCOC=LOCOC04    FLARE=FLARE02    START=START03    RLFD=RLFD02    YETA=YETA01    PSIE=PSIE01
GSFD=GSFD02      .NOT.CFDIN1    TMEFLG.NE.0    .NOT.CCRUFZD    $AC1=TP2      $AC2=N($AC1-TP1)
$M01=LIM180      $AC3=SRLMT($AC2,$M01)    $AC4=$AC3*2**3    $AC5=-$AC4    LOCSTR1=$AC5    $AC6=YETA
$M02=MULTM($AC6,PSIE)    $AC7=MULT($AC6,PSIE)    $AC7.EQ.0      $AC8=LOCSTR1    $M03=MULTM($AC8,PSIE)    $AC9=MULT($AC8,PSIE)
$AC9.LT.0
.IMPLIES.
R($AC3*2**3)      CF07      LOCSTR1=LOCSTR01

```

```

--- CONTROL PATH 27 ---

*CEXIT0, SPEC, CF07, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCDF=LOCDF05, LOCOC=LOCOC04, FL
ARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT.CFDIN1, TMEFLG.NE.0, LOCSTR=LOCSTR01, .NOT.CCRUFZD*

LDA SET      STA CAPT      LDA HRAD      SUB A1200      %R($AC-A1200)X      .NOT.($AC.LT.0)
LDA LOCSTR   LDQ DEG25      TRA SRLMT      SUB RLFD      TRA SRLMT      MPY DELTAT
XABS($AC).LT.ABS(TAU3)X      DIV TAU3      LDQ DEG4      TRA SRLMT      STA RCMD
ADS RLFD      SUB ROLL      LDQ DEG25      TRA SRLMT      STA RCMD
*CEXIT0, SPEC, CF011, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602
TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01*

```

```

** VERIFICATION CONDITIONS **

CEXIT0      SPEC      CF07      MPILS=MPILS01      TP1=TP103      TP2=TP205
TP3=TP302    TP4=TP402    TP5=TP502    TP6=TP602    TP7=TP704    LOCDF=LOCDF06
LOCOC=LOCOC04    FLARE=FLARE02    START=START03    RLFD=RLFD02    YETA=YETA01    PSIE=PSIE01
GSFD=GSFD02      .NOT.CFDIN1    TMEFLG.NE.0    LOCSTR=LOCSTR01    $AC1=SET      $M01=DEG25
CAPT1=$AC1      $AC2=HRAD      $AC3=$AC2-A1200    .NOT.($AC3.LT.0)    $AC4=LOCSTR    $AC9=SR.MT($AC8,$M02)
$AC5=SRLMT($AC4,$M01)    $AC6=N($AC5-RLFD)    $AC7=$AC6/2**2    $AC8=$AC7/TAU3    $M02=DEG4    RLFD1=$AC11
$M03=MULTM($AC9,DELTAT)    $AC10=MULT($AC9,DELTAT)    $AC11=N($AC10+RLFD)    $AC11=$AC13
$AC12=N($AC11-ROLL)    $M04=DEG25      $AC13=SRLMT($AC12,$M04)
.IMPLIES.
R($AC2-A1200)    ABS($AC7).LT.ABS(TAU3)    CF011      START=START04      LOCDF=LOCDF06
TP1=TP104      TP2=TP206      YETA=YETA02    PSIE=PSIE02    LOCSTR=LOCSTR02

```

ORIGINAL PAGE IS
OF POOR QUALITY

CAPT1=CAPT04

RLFD1=RLFD04

RCMD1=RCMD01

--- CONTROL PATH 28 ---

CEX10, SPEC, CFD7, MPILS=MPILS01, TP1=TP103, TP2=TP205, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP704, LOCFC=LOCFC05, LOCOC=LOCOC04, FLARE=FLARE02, START=START03, RLFD=RLFD02, YETA=YETA01, PSIE=PSIE01, GSFD=GSFD02, .NOT, CFDIN1, TMEFLG, NE, 0, LOCSTR=LOCSTR01, .NOT, CCRUFZD

LDA SET	STA CAPT	LDA HRAD	SUB A1200	%R(\$AC-A1200)%	(\$AC.LT.0)
LDA LOCSTR	LDQ DEG15	TRA SRLMT	SUB RLFD	SAR 2	MPY DELTAT
XABS(\$AC).LT.ABS(TAU3)X	DIV TAU3	LDQ DEG4	TRA SRLMT	STA RCMD	
ADS RLFD	SUB ROLL	LDQ DEG25	TRA SRLMT		

CEX10, SPEC, CFD11, START=START04, LOCFC=LOCFC06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01

** VERIFICATION CONDITIONS **

CEX10
TP3=TP302
LOCOC=LOCOC04
GSFD=GSFD02
CAPT1=SAC1
\$AC5=SRLMT(\$AC4,\$MQ1)
\$MQ3=MULTM(\$AC9,DELTAT)
\$AC12=N(\$AC11-ROLL)
\$AC12=N(\$AC11-ROLL)
\$MQ4=DEG25
IMPLIES.
R(\$AC2-A1200)
TP1=TP104
CAPT1=CAPT04

SPEC
TP4=TP402
FLARE=FLARE02
NDT, CFDIN1
\$AC2=HRAD
\$AC6=N(\$AC5-RLFD)
\$MQ4=DEG25

ABS(\$AC7).LT.ABS(TAU3)
TP2=TP206
RLFD1=RLFD04

CFD7
TP5=TP502
START=START03
TMEFLG, NE, 0
\$AC3=\$AC2-A1200
\$AC7=\$AC6/2**2
\$AC10=MULT(\$AC9,DELTAT)
\$AC13=SRLMT(\$AC12,\$MQ4)

MPILS=MPILS01
TP6=TP602
RLFD=RLFD02
LOCSTR=LOCSTR01
\$AC3.LT.0
\$AC8=\$AC7/TAU3

TP1=TP103
TP7=TP704
YETA=YETA01
NDT, CCRUFZD
\$AC4=LOCSTR
\$MQ2=DEG4
\$AC11=N(\$AC10+RLFD)
RCMD1=SAC13

START=START04
PSIE=PSIE02

TP2=TP205
LOCFC=LOCFC05
PSIE=PSIE01
\$AC1=SET
\$MQ1=DEG15
\$AC9=SRLMT(\$AC8,\$MQ2)
RLFD1=SAC11

LOCFC=LOCFC06
LOCSTR=LOCSTR02

--- CONTROL PATH 29 ---

CEX10, SPEC, CCRUFZD, START=START04, LOCFC=LOCFC06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG, NE, 0, .NOT, CFD42

LDA GUID2D	.NOT, (\$AC.NE.0)	STZ RCMVLD	STZ PCMVLD
------------	-------------------	------------	------------

FALSE.

** VERIFICATION CONDITIONS **

CEX10
MPILS=MPILS01
TP6=TP602
TMEFLG, NE, 0

SPEC
TP1=TP102
TP7=TP702
NDT, CFD42

CCRUFZD
TP2=TP202
LOCOC=LOCOC04
\$AC1=GUID2D

START=START04
TP2=TP302
FLARE=FLARE02
NDT, (\$AC1.NE.0)

LOCFC=LOCFC06
TP4=TP402
YETA=YETA02
RCMVLD1=0

GSFD=GSFD02
TP5=TP502
PSIE=PSIE02
PCMVLD1=0

IMPLIES.
FALSE.

--- CONTROL PATH 30 ---

CEX10, SPEC, CCRUFZD, START=START04, LOCFC=LOCFC06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP102, TP2=TP202, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP702, LOCOC=LOCOC04, FLARE=FLARE02, YETA=YETA02, PSIE=PSIE02, TMEFLG, NE, 0, .NOT, CFD42

LDA GUID2D	(\$AC.NE.0)	LDA LATSTR	LDQ DEG25	TRA SRLMT	SUB RLFD
SAR 2	XABS(\$AC).LT.ABS(TAU3)X	DIV TAU3	LDQ DEG4	TRA SRLMT	TRA SRLMT
MPY DELTAT	ADS RLFD	SUB ROLL	LDQ DEG25	TRA SRLMT	STA RCMD

CEX10, SPEC, CFD11, START=START04, LOCFC=LOCFC06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01

** VERIFICATION CONDITIONS **

CEX10
MPILS=MPILS01
TP6=TP602
TMEFLG, NE, 0
\$AC3=SRLMT(\$AC2,\$MQ1)
\$MQ3=MULTM(\$AC7,DELTAT)
\$AC10=N(\$AC9-ROLL)
\$MQ4=DEG25

SPEC
TP1=TP102
TP7=TP702
NDT, CFD42

\$AC4=N(\$AC3-RLFD)
\$MQ4=DEG25

CCRUFZD
TP2=TP202
LOCOC=LOCOC04
\$AC1=GUID2D
\$AC5=\$AC4/2**2
\$AC8=MULT(\$AC7,DELTAT)
\$AC11=SRLMT(\$AC10,\$MQ4)

START=START04
TP3=TP302
FLARE=FLARE02
\$AC1.NE.0
\$AC6=\$AC5/TAU3

LOCFC=LOCFC06
TP4=TP402
YETA=YETA02
\$AC2=LATSTR
\$MQ2=DEG4
\$AC9=N(\$AC8+RLFD)
RCMD1=SAC11

GSFD=GSFD02
TP5=TP502
PSIE=PSIE02
\$MQ1=DEG25
\$AC7=SRLMT(\$AC6,\$MQ2)
RLFD1=SAC9

LOCSTR=LOCSTR02 CAPT=CAPT04 RLFD1=RLFD04 RCMD1=RCMD01

--- CONTROL PATH 31 ---

CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01
LDA LOCFD .NOT. (\$AC.EQ.0) LDA LOCVD .NOT. (\$AC.EQ.0) LDA MPILS .NOT. (\$AC.EQ.0)
LDA SET STA RCMVLD
CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	\$AC1=LOCFD	.NOT. (\$AC1.EQ.0)
\$AC2=LOCVD	.NOT. (\$AC2.EQ.0)	\$AC3=MPILS	.NOT. (\$AC3.EQ.0)	\$AC4=SET	RCMVLD1=\$AC4

.IMPLIES.
RCMVLD1=RCMVLD04

--- CONTROL PATH 32 ---

CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01
LDA LOCFD .NOT. (\$AC.EQ.0) LDA LOCVD .NOT. (\$AC.EQ.0) LDA MPILS (\$AC.EQ.0)
STZ RCMVLD
CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	\$AC1=LOCFD	.NOT. (\$AC1.EQ.0)
\$AC2=LOCVD	.NOT. (\$AC2.EQ.0)	\$AC3=MPILS	\$AC3.EQ.0	RCMVLD1=0	

.IMPLIES.
RCMVLD1=RCMVLD04

--- CONTROL PATH 33 ---

CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01
LDA LOCFD .NOT. (\$AC.EQ.0) LDA LOCVD (\$AC.EQ.0) STZ RCMVLD
CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	\$AC1=LOCFD	.NOT. (\$AC1.EQ.0)
\$AC2=LOCVD	\$AC2.EQ.0	RCMVLD1=0			

.IMPLIES.
RCMVLD1=RCMVLD04

--- CONTROL PATH 34 ---

CEXIT0, SPEC, CFD11, START=START04, LOCFD=LCCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01
LDA LOCFD (\$AC.EQ.0) LDA LOCVD .NOT. (\$AC.EQ.0) STZ RCMVLD

ORIGINAL PAGE IS
OF POOR QUALITY

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	SAC1=LOCDF	SAC1.EQ.0
SAC2=GUID2D	.NOT.(\$AC2.NE.0)	RCMVL01=0			
.IMPLIES.					
RCMVL01=RCMVL004					

--- CONTROL PATH 35 ---

*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL01=RCMVL004

LDA LOCDF (\$AC.EQ.0) LDA GUID2D (\$AC.NE.0) LDA SET STA RCMVL01

*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL01=RCMVL004

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	SAC1=LOCDF	SAC1.EQ.0
SAC2=GUID2D	SAC2.NE.0	SAC3=SET	RCMVL01=\$AC3		
.IMPLIES.					
RCMVL01=RCMVL004					

--- CONTROL PATH 36 ---

*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL01=RCMVL004

LDA LOCDF .NOT.(\$AC.EQ.0) LDA GSFD .NOT.(\$AC.NE.0) LDA GSBSNS .NOT.(\$AC.EQ.0)

LDA SET STA GSFD STA LGAMC LDA GSDEV MPY K1

XDIR(\$AQ*2**8)% SLL 8 SUB TP3 %ABS(\$AC).LT.ABS(TAU4)% DIV TAU4

MPY DELTAT SAR 4 ADS TP3 %ABS(\$AC).LT.ABS(C180BF)% DIV C180BF

MPY K3 ADD GSA SUB K4 CMA STA TP4

*CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL01=RCMVL004, GSFD=GSFD03, LGAMC=LGAMC02

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL01=RCMVL004	SAC1=LOCDF
.NOT.(\$AC1.EQ.0)	SAC2=GSFD	.NOT.(\$AC2.NE.0)	SAC3=GSBSNS	.NOT.(\$AC3.EQ.0)	SAC4=SET
GSFD1=\$AC4	SAC5=GAMMA	LGAMC1=\$AC5	SAC6=GSDEV	SMQ1=MULTM(\$AC6,K1)	SAC7=MULT(\$AC5,K1)
\$AQ1=(\$AC7,SMQ1)	\$AQ2=\$AQ1*2**8	SAC8=AR(\$AQ2)	SMQ2=MR(\$AQ2)	SAC9=N(\$AC8-TP3)	SAC10=\$AC9/TAU4
SMQ3=MULTM(\$AC10,DELTAT)	SAC14=\$AC13/C180BF	SAC11=MULT(\$AC10,DELTAT)	SMQ4=MULTM(\$AC14,K3)	SAC12=\$AC11/2**4	SAC13=N(\$AC12+TP3)
TP31=\$AC13	TP41=\$AC18	SAC15=MULT(\$AC14,K3)	SAC16=N(\$AC15+GSA)		SAC17=N(\$AC16-K4)
SAC18=-\$AC17					
.IMPLIES.					
DIR(\$AQ1*2**8)	ABS(\$AC9).LT.ABS(TAU4)	GSFD1=GSFD03	ABS(\$AC13).LT.ABS(C180BF)	CFD15	
TP31=TP303	TP41=TP403		LGAMC1=LGAMC02		

--- CONTROL PATH 37 ---

*CEXIT0, SPEC, CFD11, START=START04, LOCDF=LOCDF06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL01=RCMVL004

LDA LOCDF .NOT.(\$AC.EQ.0) LDA GSFD .NOT.(\$AC.NE.0) LDA GSBSNS (\$AC.EQ.0)

LDA GUID2D .NOT.(\$AC.EQ.0) LDA VERSTR 100 05015 TGA 10000


```

SUB PCMD      XABS($AC).LT.ABS(C3B2)%      DIV C3B2      MPY DELTAT      SAR 2
ADS PCMD      LDA MDA      SUB HRAD      ZR($AC-HRAD)%      .NOT.($AC.LT.0)      LDA TIMER
SUB BINTME    XR($AC-BINTME)%      ($AC.EQ.0)
*CEXIT0,PCMD=PCMD02,RCMVL=RCMVL004,RCMD=RCMD01,RLFD=RLFD04,START=START04,LOCDF=LOCDF06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP20
6,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602,TP7=TP705,FLARE=FLARE02,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
KFLR=KFLR01*

```

** VERIFICATION CONDITIONS **

```

CEXIT0      SPEC      CFD11      START=START04      LOCDF=LOCDF06      GSFD=GSFD02
MPILS=MPILS01      TP1=TP104      TP2=TP206      TP3=TP302      TP4=TP402      TP5=TP502
TP6=TP602      TP7=TP705      FLARE=FLARE02      LOCOC=LOCOC04      YETA=YETA02      PSIE=PSIE02
LOCSTR=LOCSTR02      CAPT=CAPT04      RLFD=RLFD04      RCMD=RCMD01      RCMVL=RCMVL004      SAC1=LOCDF
.NOT.($AC1.EQ.0)      $AC2=GSFD      .NOT.($AC2.NE.0)      $AC3=GSBSNS      $AC3.EQ.0      $AC4=GUID3D
.NOT.($AC4.EQ.0)      $AC5=VERSTR      $M01=C5B15      $AC6=SRLMT($AC5,$M01)      $AC7=$AC6/VGS      $AC8=$AC7/2**2
$AC9=$AC8/KHD      $AC10=$AC9/PIB2      $M02=DEG6      $AC11=SRLMT($AC10,$M02)      $AC12=N($AC11-PCMD)
$AC13=$AC12/C3B2      $M03=MULTM($AC13,DELTAT)      $AC14=MULT($AC13,DELTAT)      $AC15=$AC14/2**2
$AC16=N($AC15+PCMD)      PCMD1=$AC16      $AC17=MDA      $AC18=$AC17-HRAD      .NOT.($AC18.LT.0)      $AC19=TIMER
$AC20=$AC19-BINTME      $AC20.EQ.0
.IMPLIES.
ADS($AC6).LT.ABS(VGS)      ABS($AC8).LT.ABS(KHD)      ABS($AC9).LT.ABS(PIB2)      ABS($AC12).LT.ABS(C3B2)
R($AC17-HRAD)      R($AC19-BINTME)      PCMD1=PCMD02      KFLR=KFLR01

```

--- CONTROL PATH 38 ---

```

*CEXIT0,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL004*
LDA LOCDF      .NOT.($AC.EQ.0)      LDA GSFD      .NOT.($AC.NE.0)      LDA GSBSNS      ($AC.EQ.0)
LDA GUID3D      .NOT.($AC.EQ.0)      LDA VERSTR      LDA C5B15      TRA SRLMT
XABS($AC).LT.ABS(VGS)%      XABS($AC).LT.ABS(PIB2)%      DIV VGS      SAR 2      XABS($AC).LT.ABS(KHD)%
DIV KHD      XABS($AC).LT.ABS(C3B2)%      DIV PIB2      LDA DEG6      TRA SRLMT
SUB PCMD      LDA MDA      SUB HRAD      ZR($AC-HRAD)%      MPY DELTAT      SAR 2
ADS PCMD      XR($AC-BINTME)%      (.NOT.($AC.EQ.0))      LDA BINTME      .NOT.($AC.LT.0)      LDA TIMER
SUB BINTME      STZ PCMVL      .NOT.($AC.EQ.0)      STA TIMER      LDA PCMVL
.NOT.($AC.EQ.0)
*FALSE.*

```

** VERIFICATION CONDITIONS **

```

CEXIT0      SPEC      CFD11      START=START04      LOCDF=LOCDF06      GSFD=GSFD02
MPILS=MPILS01      TP1=TP104      TP2=TP206      TP3=TP302      TP4=TP402      TP5=TP502
TP6=TP602      TP7=TP705      FLARE=FLARE02      LOCOC=LOCOC04      YETA=YETA02      PSIE=PSIE02
LOCSTR=LOCSTR02      CAPT=CAPT04      RLFD=RLFD04      RCMD=RCMD01      RCMVL=RCMVL004      SAC1=LOCDF
.NOT.($AC1.EQ.0)      $AC2=GSFD      .NOT.($AC2.NE.0)      $AC3=GSBSNS      $AC3.EQ.0      $AC4=GUID3D
.NOT.($AC4.EQ.0)      $AC5=VERSTR      $M01=C5B15      $AC6=SRLMT($AC5,$M01)      $AC7=$AC6/VGS      $AC8=$AC7/2**2
$AC9=$AC8/KHD      $AC10=$AC9/PIB2      $M02=DEG6      $AC11=SRLMT($AC10,$M02)      $AC12=N($AC11-PCMD)
$AC13=$AC12/C3B2      $M03=MULTM($AC13,DELTAT)      $AC14=MULT($AC13,DELTAT)      $AC15=$AC14/2**2
$AC16=N($AC15+PCMD)      PCMD1=$AC16      $AC17=MDA      $AC18=$AC17-HRAD      .NOT.($AC18.LT.0)      $AC19=TIMER
$AC20=$AC19-BINTME      .NOT.($AC20.EQ.0)      $AC21=BINTME      $AC22=PCMVL      .NOT.($AC22.EQ.0)
PCMVL1=0
.IMPLIES.
ABS($AC6).LT.ABS(VGS)      ABS($AC8).LT.ABS(KHD)      ABS($AC9).LT.ABS(PIB2)      ABS($AC12).LT.ABS(C3B2)
R($AC17-HRAD)      R($AC19-BINTME)      .FALSE.

```

--- CONTROL PATH 39 ---

```

*CEXIT0,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL004*
LDA LOCDF      .NOT.($AC.EQ.0)      LDA GSFD      .NOT.($AC.NE.0)      LDA GSBSNS      ($AC.EQ.0)
LDA GUID3D      .NOT.($AC.EQ.0)      LDA VERSTR      LDA C5B15      TRA SRLMT
XABS($AC).LT.ABS(VGS)%      XABS($AC).LT.ABS(PIB2)%      DIV VGS      SAR 2      XABS($AC).LT.ABS(KHD)%
DIV KHD      XABS($AC).LT.ABS(C3B2)%      DIV PIB2      LDA DEG6      TRA SRLMT
SUB PCMD      LDA MDA      SUB HRAD      ZR($AC-HRAD)%      MPY DELTAT      SAR 2
ADS PCMD      XR($AC-BINTME)%      (.NOT.($AC.EQ.0))      LDA BINTME      .NOT.($AC.LT.0)      LDA TIMER
.NOT.($AC.EQ.0)

```

ORIGINAL PAGE IS
OF POOR QUALITY

.FALSE.

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCDC=LOCDC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	SAC1=LOCDF
.NOT.(\$AC1.EQ.0)	SAC2=GSFD	.NOT.(\$AC2.NE.0)	SAC3=GSBSNS	SAC3.EQ.0	SAC4=GUID30
.NOT.(\$AC4.EQ.0)	SAC5=VERSTR	\$MQ1=C5B15	SAC6=SRLMT(\$AC5,\$MQ1)	SAC7=\$AC6/VGS	SAC8=\$AC7/2**2
\$AC9=\$AC8/KHD	SAC10=\$AC9/PIB2	\$MQ2=DEG6	SAC11=SRLMT(\$AC10,\$MQ2)		SAC12=N(\$AC11-PCMD)
\$AC13=\$AC12/C3B2	\$MQ3=MULT(\$AC13,DELTAT)		SAC14=MULT(\$AC13,DELTAT)		SAC15=\$AC14/2**2
\$AC16=N(\$AC15+PCMD)	PCMD1=\$AC16	SAC17=MDA	SAC18=\$AC17-HRAD	.NOT.(\$AC18.LT.0)	SAC19=TIMER
\$AC20=\$AC19-BINTME	.NOT.(\$AC20.EQ.0)	SAC21=BINTME	TIME1=\$AC21	SAC22=PCMVLD	SAC22.EQ.0
\$AC23=RCMVLD	PCMVLD1=\$AC23				
.IMPLIES.					
ABS(\$AC6).LT.ABS(VGS)	ABS(\$AC8).LT.ABS(KHD)	ABS(\$AC9).LT.ABS(PIB2)		ABS(\$AC12).LT.ABS(C3B2)	
R(\$AC17-HRAD)	R(\$AC19-BINTME)	.FALSE.			

--- CONTROL PATH 40 ---

*CEXIT0,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
TP7=TP705,FLARE=FLARE02,LOCDC=LOCDC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04*

LDA LOCDF	.NOT.(\$AC.EQ.0)	LDA GSFD	.NOT.(\$AC.NE.0)	LDA GSBSNS	(\$AC.EQ.0)
LDA GUID30	.NOT.(\$AC.EQ.0)	LDA VERSTR	LDO C5B15	TRA SRLMT	
XABS(\$AC).LT.ABS(VGS)		DIV VGS	SAR 2	XABS(\$AC).LT.ABS(KHD)	X
DIV KHD	XABS(\$AC).LT.ABS(PIB2)		DIV PIB2	LDO DEG6	TRA
SUB PCMD	XABS(\$AC).LT.ABS(C3B2)		DIV C3B2	MPY DELTAT	SAR 2
ADS PCMD	LDA MDA	SUB HRAD	XR(\$AC-HRAD)	(\$AC.LT.0)	LDA RCMVL
STA PCMVLD					

.FALSE.

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCDC=LOCDC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	SAC1=LOCDF
.NOT.(\$AC1.EQ.0)	SAC2=GSFD	.NOT.(\$AC2.NE.0)	SAC3=GSBSNS	SAC3.EQ.0	SAC4=GUID30
.NOT.(\$AC4.EQ.0)	SAC5=VERSTR	\$MQ1=C5B15	SAC6=SRLMT(\$AC5,\$MQ1)	SAC7=\$AC6/VGS	SAC8=\$AC7/2**2
\$AC9=\$AC8/KHD	SAC10=\$AC9/PIB2	\$MQ2=DEG6	SAC11=SRLMT(\$AC10,\$MQ2)		SAC12=N(\$AC11-PCMD)
\$AC13=\$AC12/C3B2	\$MQ3=MULT(\$AC13,DELTAT)		SAC14=MULT(\$AC13,DELTAT)		SAC15=\$AC14/2**2
\$AC16=N(\$AC15+PCMD)	PCMD1=\$AC16	SAC17=MDA	SAC18=\$AC17-HRAD	SAC18.LT.0	SAC19=RCMVLD
PCMVLD1=\$AC19					
.IMPLIES.					
ABS(\$AC6).LT.ABS(VGS)	ABS(\$AC8).LT.ABS(KHD)	ABS(\$AC9).LT.ABS(PIB2)		ABS(\$AC12).LT.ABS(C3B2)	
R(\$AC17-HRAD)	.FALSE.				

--- CONTROL PATH 41 ---

*CEXIT0,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
TP7=TP705,FLARE=FLARE02,LOCDC=LOCDC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04*

LDA LOCDF	.NOT.(\$AC.EQ.0)	LDA GSFD	.NOT.(\$AC.NE.0)	LDA GSBSNS	(\$AC.EQ.0)
LDA GUID30	(\$AC.EQ.0)	STZ PCMVLD			

.FALSE.

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCDF=LOCDF06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCDC=LOCDC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	SAC1=LOCDF
.NOT.(\$AC1.EQ.0)	SAC2=GSFD	.NOT.(\$AC2.NE.0)	SAC3=GSBSNS	SAC3.EQ.0	SAC4=GUID30
\$AC4.EQ.0	PCMVLD1=0				

--- CONTROL PATH 42 ---

```
*CEXITO,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04*
LDA LOCDF      .NOT.($AC.EQ.0)      LDA GSFD      ($AC.NE.0)      LDA GSDEV      MPY K1
%DIR($AQ*2**8)X  SLL 8              SUB TP3        %ABS($AC).LT.ABS(TAU4)X  DIV TAU4
MPY DELTAT      SAR 4              ADS TP3        %ABS($AC).LT.ABS(C180BF)X  DIV C180BF
MPY K3          ADD GSA            SUB K4          CMA          STA TP4
*CEXITC,SPEC,CFD15,START=START04,LOCDF=LCCFD06,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP303,TP4=TP403,TP5=TP502,TP6=TP602,TP7=TP705,F
LARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04,GSFD=GSFD03,L
GAMC=LGAMC02*
```

** VERIFICATION CONDITIONS **

CEXITO	SPEC	CFD11	START=START04	LOCDF=LCCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LCCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	\$AC1=LOCDF
.NOT.(\$AC1.EQ.0)	\$AC2=GSFD	\$AC2.NE.0	\$AC3=GSDEV	\$M01=MULTM(\$AC3,K1)	\$AC4=MULT(\$AC3,K1)
\$A01=(\$AC4,\$M01)	\$AQ2=\$A01*2**8	\$AC5=AR(\$AQ2)	\$M02=MR(\$AQ2)	\$AC6=N(\$AC5-TP3)	\$AC7=\$AC6/TAU4
\$M03=MULTM(\$AC7,DELTAT)	\$AC11=\$AC10/C180BF	\$AC8=MULT(\$AC7,DELTAT)	\$AC12=MULT(\$AC11,K3)	\$AC9=\$AC8/2**4	\$AC10=N(\$AC9+TP3)
TP31=\$AC10	TP41=\$AC15	\$M04=MULTM(\$AC11,K3)	\$AC13=N(\$AC12+GSA)		\$AC14=N(\$AC13-K4)
\$AC15=\$AC14					
.IMPLIES.					
DIR(\$A01*2**8)	ABS(\$AC6).LT.ABS(TAU4)		ABS(\$AC10).LT.ABS(C180BF)		CFD15
TP31=TP303	TP41=TP403	GSFD=GSFD03	LGAMC=LGAMC02		

--- CONTROL PATH 43 ---

```
*CEXITO,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04*
LDA LOCDF      ($AC.EQ.0)      LDA GUID3D      .NOT.($AC.EQ.0)      LDA VERSTR      LDQ C5B15
TRA SRLMT      %ABS($AC).LT.ABS(VGS)X  DIV VGS      SAR 2
%ABS($AC).LT.ABS(KHD)X  DIV KHD      %ABS($AC).LT.ABS(PIB2)X  DIV PIB2
LDQ DEG6      TRA SRLMT      SUB PCMD      %ABS($AC).LT.ABS(C3B2)X  DIV C3B2
MPY DELTAT      SAR 2          ADS PCMD      LDA MDA      SUB HRAD      %R($AC-HRAD)X
.NOT.($AC.LT.0)      LDA TIMER      SUB QINTME      %R($AC-BINTME)X  ($AC.EQ.0)
*CEXITO,PCMD=PCMD02,RCMVL=RCMVL04,RCMD=RCMD01,RLFD=RLFD04,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP20
6,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602,TP7=TP705,FLARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
KFLR=KFLR01*
```

** VERIFICATION CONDITIONS **

CEXITO	SPEC	CFD11	START=START04	LOCDF=LCCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LCCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	\$AC1=LOCDF
\$AC1.EQ.0	\$AC2=GUID3D	.NOT.(\$AC2.EQ.0)	\$AC3=VERSTR	\$M01=C5B15	\$AC4=SRLMT(\$AC3,\$M01)
\$AC5=\$AC4/VGS	\$AC6=\$AC5/2**2	\$AC7=\$AC6/KHD	\$AC8=\$AC7/PIB2	\$M02=DEG6	\$AC9=SRLMT(\$AC8,\$M02)
\$AC10=N(\$AC9-PCMD)	\$AC11=\$AC10/C3B2	\$M03=MULTM(\$AC11,DELTAT)	\$AC15=MDA	\$AC12=MULT(\$AC11,DELTAT)	
\$AC13=\$AC12/2**2	\$AC14=N(\$AC13+PCMD)	PCMD1=\$AC14	\$AC18=EQ.0	\$AC16=\$AC15-HRAD	.NOT.(\$AC15.LT.0)
\$AC17=TIMER	\$AC18=\$AC17-BINTME				
.IMPLIES.					
ABS(\$AC4).LT.ABS(VGS)	ABS(\$AC6).LT.ABS(KHD)	ABS(\$AC7).LT.ABS(PIB2)		ABS(\$AC10).LT.ABS(C3B2)	
R(\$AC15-HRAD)	R(\$AC17-BINTME)	PCMD1=PCMD02	KFLR=KFLR01		

--- CONTROL PATH 44 ---

```
*CEXITO,SPEC,CFD11,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVL=RCMVL04*
LDA LOCDF      ($AC.EQ.0)      LDA GUID3D      .NOT.($AC.EQ.0)      LDA VERSTR      LDQ C5B15
TRA SRLMT      %ABS($AC).LT.ABS(VGS)X  DIV VGS      SAR 2
%ABS($AC).LT.ABS(KHD)X  DIV KHD      %ABS($AC).LT.ABS(PIB2)X  DIV PIB2
LDQ DEG6      TRA SRLMT      SUB PCMD      %ABS($AC).LT.ABS(C3B2)X  DIV C3B2
MPY DELTAT      SAR 2          ADS PCMD      LDA MDA      SUB HRAD      %R($AC-HRAD)X
.NOT.($AC.LT.0)      LDA TIMER      SUB QINTME      %R($AC-BINTME)X  ($AC.EQ.0)
*CEXITO,PCMD=PCMD02,RCMVL=RCMVL04,RCMD=RCMD01,RLFD=RLFD04,START=START04,LOCDF=LCCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP20
6,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602,TP7=TP705,FLARE=FLARE02,LOCOC=LCCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,
KFLR=KFLR01*
```

```

.NOT.($AC.LT.0) LDA TIMER SUB BINTME %R($AC-BINTME)% (.NOT.($AC.EQ.0)) LDA BINTME
STA TIMER LDA PCMVLD .NOT.($AC.EQ.0) STZ PCMVLD
**FALSE**

```

** VERIFICATION CONDITIONS **

CEXITO	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	SAC1=LOCFD
SAC1.EQ.0	SAC2=GUID3D	.NOT.(\$AC2.EQ.0)	SAC3=VERSTR	SMQ1=C5B15	SAC4=SRLMT(\$AC3,SMQ1)
SAC5=\$AC4/VGS	SAC6=\$AC5/2**2	SAC7=\$AC6/KHD	SAC8=\$AC7/PIB2	SMQ2=DEG6	SAC9=SRLMT(\$AC8,SMQ2)
SAC10=N(\$AC9-PCMD)	SAC11=\$AC10/C3B2	SMQ3=MULTM(\$AC11,DELTAT)		SAC12=MULT(\$AC11,DELTAT)	
SAC13=\$AC12/2**2	SAC14=N(\$AC13+PCMD)	SAC15=MDA		SAC16=\$AC15-HRAD	.NOT.(\$AC16.LT.0)
SAC17=TIMER	SAC18=\$AC17-BINTME	SAC19=BINTME		TIMER1=\$AC19	SAC20=PCMVLD
.NOT.(\$AC20.EQ.0)	PCMVLD1=0				
.IMPLIES.					
ABS(\$AC4).LT.ABS(VGS)	ABS(\$AC6).LT.ABS(KHD)	ABS(\$AC7).LT.ABS(PIB2)		ABS(\$AC10).LT.ABS(C3B2)	
R(\$AC15-HRAD)	R(\$AC17-BINTME)	.FALSE.			

--- CONTROL PATH 45 ---

```

*CEXITO,SPEC,CFD11,START=START04,LOCFD=LOCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVLD=RCMVLD04*
LDA LOCFD ($AC.EQ.0) LDA GUID3D .NOT.($AC.EQ.0) LDA VERSTR LDQ C5B15
TRA SRLMT %ABS($AC).LT.ABS(VGS)% DIV VGS SAR 2
%ABS($AC).LT.ABS(KHD)% DIV KHD %ABS($AC).LT.ABS(PIB2)% DIV PIB2
LDQ DEG6 TRA SRLMT SUS PCMD %ABS($AC).LT.ABS(C3B2)% DIV C3B2
MPY DELTAT SAR 2 ADS PCMD LDA MDA SUB HRAD %R($AC-HRAD)%
.NOT.($AC.LT.0) LDA TIMER SUB BINTME %R($AC-BINTME)% (.NOT.($AC.EQ.0)) LDA BINTME
STA TIMER LDA PCMVLD ($AC.EQ.0) LDA RCMVLD STA PCMVLD
**FALSE**

```

** VERIFICATION CONDITIONS **

CEXITO	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	SAC1=LOCFD
SAC1.EQ.0	SAC2=GUID3D	.NOT.(\$AC2.EQ.0)	SAC3=VERSTR	SMQ1=C5B15	SAC4=SRLMT(\$AC3,SMQ1)
SAC5=\$AC4/VGS	SAC6=\$AC5/2**2	SAC7=\$AC6/KHD	SAC8=\$AC7/PIB2	SMQ2=DEG6	SAC9=SRLMT(\$AC8,SMQ2)
SAC10=N(\$AC9-PCMD)	SAC11=\$AC10/C3B2	SMQ3=MULTM(\$AC11,DELTAT)		SAC12=MULT(\$AC11,DELTAT)	
SAC13=\$AC12/2**2	SAC14=N(\$AC13+PCMD)	SAC15=MDA		SAC16=\$AC15-HRAD	.NOT.(\$AC16.LT.0)
SAC17=TIMER	SAC18=\$AC17-BINTME	SAC19=BINTME		TIMER1=\$AC19	SAC20=PCMVLD
SAC20.EQ.0	SAC21=RCMVLD	PCMVLD1=\$AC21			
.IMPLIES.					
ABS(\$AC4).LT.ABS(VGS)	ABS(\$AC6).LT.ABS(KHD)	ABS(\$AC7).LT.ABS(PIB2)		ABS(\$AC10).LT.ABS(C3B2)	
R(\$AC15-HRAD)	R(\$AC17-BINTME)	.FALSE.			

--- CONTROL PATH 46 ---

```

*CEXITO,SPEC,CFD11,START=START04,LOCFD=LOCFD06,GSFD=GSFD02,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP302,TP4=TP402,TP5=TP502,TP6=TP602
,TP7=TP705,FLARE=FLARE02,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVLD=RCMVLD04*
LDA LOCFD ($AC.EQ.0) LDA GUID3D .NOT.($AC.EQ.0) LDA VERSTR LDQ C5B15
TRA SRLMT %ABS($AC).LT.ABS(VGS)% DIV VGS SAR 2
%ABS($AC).LT.ABS(KHD)% DIV KHD %ABS($AC).LT.ABS(PIB2)% DIV PIB2
LDQ DEG6 TRA SRLMT SUB PCMD %ABS($AC).LT.ABS(C3B2)% DIV C3B2
MPY DELTAT SAR 2 ADS PCMD LDA MDA SUB HRAD %R($AC-HRAD)%
($AC.LT.0) LDA RCMVLD STA PCMVLD
**FALSE**

```

** VERIFICATION CONDITIONS **

CEXITO	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104				

ORIGINAL PAGE IS
OF POOR QUALITY

LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	\$AC1=LOCFO
\$AC1=EQ.0	\$AC2=GUID3D	.NOT.(\$AC2=EQ.0)	\$AC3=VERSTR	\$MQ1=C5B15	\$AC4=SRLMT(\$AC3,\$MQ1)
\$AC5=\$AC4/VGS	\$AC6=\$AC5/2**2	\$AC7=\$AC6/KHD	\$AC8=\$AC7/PIB2	\$MQ2=DEG6	\$AC9=SRMT(\$AC8,\$MQ2)
\$AC10=N(\$AC9-PCMD)	\$AC11=\$AC10/C3B2	\$MQ3=MULTM(\$AC11,DELTAT)		\$AC12=MULT(\$AC11,DELTAT)	
\$AC13=\$AC12/2**2	\$AC14=N(\$AC13+PCMD)	PCMD1=\$AC14	\$AC15=MDA	\$AC16=\$AC15-HRAD	\$AC16.LT.0
\$AC17=RCMVLD	PCMVLD1=\$AC17				
.IMPLIES.					
ABS(\$AC4).LT.ABS(VGS)	ABS(\$AC6).LT.ABS(KHD)	ABS(\$AC7).LT.ABS(PIB2)		ABS(\$AC10).LT.ABS(C3B2)	
R(\$AC15-HRAD)	.FALSE.				

--- CONTROL PATH 47 ---

*CEXIT0, SPEC, CFD11, START=START04, LOCFD=LOCFD06, GSFD=GSFD02, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP302, TP4=TP402, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

LDA LOCFD (\$AC.EQ.0) LDA GUID3D (\$AC.EQ.0) STZ PCMVLD

** FALSE **

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD11	START=START04	LOCFD=LOCFD06	GSFD=GSFD02
MPILS=MPILS01	TP1=TP104	TP2=TP206	TP3=TP302	TP4=TP402	TP5=TP502
TP6=TP602	TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02
LOCSTR=LOCSTR02	CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	\$AC1=LOCFO
\$AC1=EQ.0	\$AC2=GUID3D	\$AC2=EQ.0	PCMVLD1=0		
.IMPLIES.					
.FALSE.					

--- CONTROL PATH 48 ---

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

LDA FLARE .NOT.(\$AC.NE.0) LDA HDCF ADD K2 XR(\$AC+K2)X .NOT.(\$AC.LT.0)

LDA K10 ADD K7 XR(\$AC+K7)X MPY TF XABS(\$AC).LT.ABS(K8)X

DIV K8 STA HREF

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCFD=LOCFD06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602
TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	GSFD=GSFD03	LGAMC=LGAMC02
\$AC1=FLARE	.NOT.(\$AC1.NE.0)	\$AC2=HDCF	\$AC3=\$AC2+K2	.NOT.(\$AC3.LT.0)	\$AC4=K10
\$AC5=\$AC4+K7	\$MQ1=MULTM(\$AC5,TF)	\$AC6=MULT(\$AC5,TF)	\$AC7=\$AC6/K8	HREF1=\$AC7	
.IMPLIES.					
R(\$AC2+K2)	R(\$AC4+K7)	ABS(\$AC6).LT.ABS(K8)	HREF1=HREF02		

--- CONTROL PATH 49 ---

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

LDA FLARE .NOT.(\$AC.NE.0) LDA HDCF ADD K2 XR(\$AC+K2)X (\$AC.LT.0)

LDA HDCF ADD K7 XR(\$AC+K7)X MPY TF XABS(\$AC).LT.ABS(K8)X

DIV K8 STA HREF

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

** VERIFICATION CONDITIONS **

TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	GSFD=GSFD03	LGAMC=LGAMC02
\$AC1=FLARE	.NDT.(\$AC1.NE.0)	\$AC7=HDCF	\$AC3=\$AC2+K2	\$AC3.LT.0	\$AC4=HDCF
\$AC5=\$AC4+K7	\$M01=MULTM(\$AC5,TF)	\$AC6=MULT(\$AC5,TF)	\$AC7=\$AC6/K8	HREF1=\$AC7	
.IMPLIES.					
R(\$AC2+K2)	R(\$AC4+K7)	ABS(\$AC6).LT.ABS(K8)	HREF1=HREF02		

--- CONTROL PATH 50 ---

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02

LDA FLARE (\$AC.NE.0)

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCDF=LOCDF06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602
TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	GSFD=GSFD03	LGAMC=LGAMC02
\$AC1=FLARE	\$AC1.NE.0				
.IMPLIES.					
HREF=HREF02					

--- CONTROL PATH 51 ---

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

LDA HRAD XABS(\$AC).LT.ABS(HREF)X DIV HREF STA KFLR SUB MSKL15

XR(\$AC-MSKL15)X .NDT.(\$AC.GE.0) LDA SET STA FLARE

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02, KFLR=KFLR01

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCDF=LOCDF06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602
TP7=TP705	FLARE=FLARE02	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVL=RCMVL04	GSFD=GSFD03	LGAMC=LGAMC02
HREF=HREF02	\$AC1=HRAD	\$AC2=\$AC1/HREF	KFLR1=\$AC2	\$AC3=\$AC2-MSKL15	.NDT.(\$AC3.GE.0)
\$AC4=SET	FLARE1=\$AC4				
.IMPLIES.					
ABS(\$AC1).LT.ABS(HREF)		R(\$AC2-MSKL15)	FLARE1=FLARE03	KFLR1=KFLR01	

--- CONTROL PATH 52 ---

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02

LDA HRAD XABS(\$AC).LT.ABS(HREF)X DIV HREF STA KFLR SUB MSKL15

XR(\$AC-MSKL15)X (\$AC.GE.0) STZ FLARE

CEXIT0, SPEC, CFD15, START=START04, LOCDF=LOCDF06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, FLARE=FLARE02, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVL=RCMVL04, GSFD=GSFD03, LGAMC=LGAMC02, HREF=HREF02, KFLR=KFLR01

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCDF=LOCDF06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602

ORIGINAL PAGE IS
OF POOR QUALITY

HREF=HREF02 SAC1=HRAD SAC2=SAC1/HREF KFLR1=SAC2 SAC3=SAC2-MSKL15 SAC3.GE.0
 FLARE1=0
 .IMPLIES.
 ABS(SAC1).LT.ABS(HREF) R(SAC2-MSKL15) FLARE1=FLARE03 KFLR1=KFLR01

--- CONTROL PATH 53 ---

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, F
 LARE=FLARE03, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GS=D=GSFD03, L
 GAMC=LGAMC02, HREF=HREF02, KFLR=KFLR01*

LDA	KFLR	LDD	MSKL15	TRA	SRLMT	%R(SAC*2**8)%	SAL	8	MPY	TP4	
SUB	K4	.NOT.	(SAC.LT.0)	LDA	A0	SUB	LGAMC	%ABS(SAC).LT.ABS(TAUS)%			
DIV	TAUS	SAR	3	LDD	LIMPR	TRA	SRLMT	MPY	DELTAT	ADS	LGAMC
SUB	GAMMA	STA	GAMCMD								

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, F
 LARE=FLARE03, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GS=D=GSFD03, G
 AMCMD=GAMCMD01, HREF=HREF02, KFLR=KFLR01, GAMMAC=GAMMAC01, LGAMC=LGAMC03*

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCFD=LOCFD06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602
TP7=TP705	FLARE=FLARE03	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	GSFD=GSFD03	LGAMC=LGAMC02
HREF=HREF02	KFLR=KFLR01	SAC1=KFLR	\$M01=MSKL15	SAC2=SRLMT(SAC1,\$M01)	SAC3=SAC2*2**8
\$M02=MULTM(SAC3,TP4)	SAC4=MULT(SAC3,TP4)	SAC5=N(SAC4-K4)	.NOT.(SAC5.LT.0)	SAC6=A0	SAC7=N(SAC6-LGAMC)
SAC8=SAC7/TAUS	SAC9=SAC8/2**3	\$M03=LIMPR	SAC10=SRLMT(SAC9,\$M03)	SAC12=N(SAC11+LGAMC)	LGAMC1=SAC12
\$M04=MULTM(SAC10,DELTAT)		SAC11=MULT(SAC10,DELTAT)			
SAC13=N(SAC12-GAMMA)	GAMCMD1=SAC13				
.IMPLIES.					
R(SAC2*2**8)	ABS(SAC7).LT.ABS(TAUS)		GAMCMD1=GAMCMD01	GAMMAC=GAMMAC01	LGAMC1=LGAMC03

--- CONTROL PATH 54 ---

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, F
 LARE=FLARE03, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GS=D=GSFD03, L
 GAMC=LGAMC02, HREF=HREF02, KFLR=KFLR01*

LDA	KFLR	LDD	MSKL15	TRA	SRLMT	%R(SAC*2**8)%	SAL	8	MPY	TP4	
SUB	K4	(SAC.LT.0)	STA	GAMMAC	ADD	GSA	ADD	GSA	.NOT.(SAC.LT.0)		
LDA	GAMMAC	SUB	LGAMC	%ABS(SAC).LT.ABS(TAUS)%			DIV	TAUS	SAR	3	
LDD	LIMPR	TRA	SRLMT	MPY	DELTAT	ADS	LGAMC	SUB	GAMMA	STA	GAMCMD

*CEXIT0, SPEC, CFD15, START=START04, LOCFD=LOCFD06, MPILS=MPILS01, TP1=TP104, TP2=TP206, TP3=TP303, TP4=TP403, TP5=TP502, TP6=TP602, TP7=TP705, F
 LARE=FLARE03, LOCOC=LOCOC04, YETA=YETA02, PSIE=PSIE02, LOCSTR=LOCSTR02, CAPT=CAPT04, RLFD=RLFD04, RCMD=RCMD01, RCMVLD=RCMVLD04, GS=D=GSFD03, G
 AMCMD=GAMCMD01, HREF=HREF02, KFLR=KFLR01, GAMMAC=GAMMAC01, LGAMC=LGAMC03*

** VERIFICATION CONDITIONS **

CEXIT0	SPEC	CFD15	START=START04	LOCFD=LOCFD06	MPILS=MPILS01
TP1=TP104	TP2=TP206	TP3=TP303	TP4=TP403	TP5=TP502	TP6=TP602
TP7=TP705	FLARE=FLARE03	LOCOC=LOCOC04	YETA=YETA02	PSIE=PSIE02	LOCSTR=LOCSTR02
CAPT=CAPT04	RLFD=RLFD04	RCMD=RCMD01	RCMVLD=RCMVLD04	GSFD=GSFD03	LGAMC=LGAMC02
HREF=HREF02	KFLR=KFLR01	SAC1=KFLR	\$M01=MSKL15	SAC2=SRLMT(SAC1,\$M01)	SAC3=SAC2*2**8
\$M02=MULTM(SAC3,TP4)	SAC4=MULT(SAC3,TP4)	SAC5=N(SAC4-K4)	SAC5.LT.0	GAMMAC1=SAC5	SAC6=N(SAC5+GSA)
SAC7=N(SAC6+GSA)	.NOT.(SAC7.LT.0)	SAC8=GAMMAC1	SAC9=N(SAC8-LGAMC)	SAC10=SAC9/TAUS	SAC11=SAC10/2**3
\$M03=LIMPR	SAC12=SRLMT(SAC11,\$M03)	SAC14=N(SAC13+LGAMC)	\$M04=MULTM(SAC12,DELTAT)		
SAC13=MULT(SAC12,DELTAT)			LGAMC1=SAC14	SAC15=N(SAC14-GAMMA)	GAMCMD1=SAC15
.IMPLIES.					
R(SAC2*2**8)	ABS(SAC9).LT.ABS(TAUS)		GAMCMD1=GAMCMD01	GAMMAC1=GAMMAC01	LGAMC1=LGAMC03

--- CONTROL PATH 55 ---

ORIGINAL PAGE IS
 OF POOR QUALITY

CAPT=CAPT04
HREF=HREF02
*NOT.(SAC1.EQ.0)
*IMPLIES.
*FALSE.

RLFD=RLFD04
GAMMAC=GAMMAC01
SAC2=RCMVLD

RCMD=RCMD01
PCMD=PCMD01
PCMVLD1=SAC2

RCMVLD=RCMVLD04
KFLR=KFLR01

GSFD=GSFD03
GAMCMD=GAMCMD01

LGAMC=LGAMC03
SAC1=GSVLD

--- CONTROL PATH 53 ---

*CEXIT0.SPEC.CFD15.START=START04,LOCFO=LOCFO06,MPILS=MPILS01,TP1=TP104,TP2=TP206,TP3=TP303,TP4=TP403,TP5=TP503,TP6=TP603,TP7=TP705.F
LARE=FLARE03,LOCOC=LOCOC04,YETA=YETA02,PSIE=PSIE02,LOCSTR=LOCSTR02,CAPT=CAPT04,RLFD=RLFD04,RCMD=RCMD01,RCMVLD=RCMVLD04,GSFD=GSFD03.L
GAMC=LGAMC03,HREF=HREF02,GAMMAC=GAMMAC01,PCMD=PCMD01,KFLR=KFLR01,GAMCMD=GAMCMD01*
LOA GSVLD (SAC.EQ.0) STZ PCMVLD
.FALSE.

** VERIFICATION CONDITIONS **

CEXIT0
TP1=TP104
TP7=TP705
CAPT=CAPT04
HREF=HREF02
SAC1.EQ.0
*IMPLIES.
*FALSE.

SPEC
TP2=TP206
FLARE=FLARE03
RLFD=RLFD04
GAMMAC=GAMMAC01
PCMVLD1=0

CFD15
TP3=TP303
LOCOC=LOCOC04
RCMD=RCMD01
PCMD=PCMD01

START=START04
TP4=TP403
YETA=YETA02
RCMVLD=RCMVLD04
KFLR=KFLR01

LOCFO=LOCFO06
TP5=TP503
PSIE=PSIE02
GSFD=GSFD03
GAMCMD=GAMCMD01

MPILS=MPILS01
TP6=TP603
LOCSTR=LOCSTR02
LGAMC=LGAMC03
SAC1=GSVLD

--- THEOREM LIBRARY ---

*****TYPE:INT,THM,RL,SET
 *****INT:I,J,K,L,M,N,P,Q,R,O
 *****THM:A,B,C,D,E
 *****RL:D,P,Q,R,U,V,W,X,Y,Z
 *****SET:S,T
 *****THE START IN FRONT OF A THEOREM INDICATES THE THEOREM CAN BE
 *****PROVED BY USING OTHER BASIC THEOREMS.
 *****THERE ARE 202 THEOREMS:
 *****54 THEOREMS CAN BE PROVED BY USING OTHER BASIC THEOREMS.
 ***** ABBR1(P,Q)=(P.EQ.Q) AND ABBR2(A,B)=(A.EQV.B)
 ***** CAN BE USED AS THEOREMS; IT MEANS THAT P IS THE ABBREVIATION OF
 ***** ARITHMETIC EXPRESSION Q OR A IS THE ABBREVIATION OF BOOLEAN
 ***** EXPRESSION B(P AND A SHOULD BE IDENTIFIERS DIFFERENT FROM ANY
 ***** VARIABLE USED IN THE CONTROL PATH TO BE PROVED).

ABBREVIATIONS

ABBR1(P,Q)=(P.EQ.Q)
 ABBR2(A,B)=(A.EQV.B)

MODULO

MOD1(I,J)=J.GT.O.I.GE.O.IMP.MOD(I,J).GE.O
 MOD2(I,J)=J.GT.O.I.GE.O.IMP.MOD(I,J).LT.J
 MOD3(I,J)=J.GT.O.I.GE.O.IMP.MOD(I,J).EQ.I-I/J
 MOD4(I,J)=J.GT.O.I.EQ.J.IMP.MOD(I,J).EQ.O
 MOD5(I)=I.GT.O.IMP.MOD(O,I).EQ.O
 MOD6(I,J,K)=MOD(I,J).EQ.O.K.EQ.J.IMP.MOD(I+K,J).EQ.O
 MOD7(I,J)=I.GE.O.J.GT.O.IMP.MOD(I+J,J).EQ.O

ABSOLUTE ARITHMETIC

ABS1(P)=ABS(P).GE.P
 ABS2(P,Q)=ABS(P-Q).LE.ABS(P)+ABS(Q)
 ABS3(P,Q)=ABS(P+Q).LE.ABS(P)+ABS(Q)
 ABS4(P,Q)=ABS(P)-ABS(Q).LE.ABS(P)+ABS(Q)

INEQUALITY AND ARITHMETIC

*AEQGE1(P,Q,R,O)=P.GE.Q.R.EQ.O.IMP.P+R.GE.O+O
 *AEQGE2(P,Q,R,O)=P.GE.Q.R.EQ.O.IMP.R+P.GE.O+O
 *AEGT3(P,Q,R,O)=P.GT.Q.R.EQ.O.IMP.P+R.GT.O+O
 *ADLT1(I,J)=I.LT.J.IMP.(I+J)/2.LT.J
 *ADLT1(I,J)=I.LT.J.IMP.(I+J)/2.LT.J
 *ADLELT1(I,J)=I.LT.J.IMP.I.LE.(I+J)/2
 *AGE1(P,Q,R,O)=P.GE.Q.R.GE.O.IMP.P+R.GE.O+O
 *AGE2(P,Q,R,O)=P.GT.Q.R.GE.O.IMP.P+R.GT.O+O
 *AGT1(P,Q,R,O)=P.GT.Q.R.GT.O.IMP.P+R.GT.O+O
 *AGT2(P,Q,R)=P.GT.Q.EQV.P+R.GT.O+R
 *AGT3(P,Q,R)=P.GT.Q.EQV.R+P.GT.O+Q
 *AGEGT1(I,J)=I.GE.J.EQV.I+1.GT.J
 *ALELT1(I,J)=I.LT.J.EQV.I+1.LE.J
 *ALT1(P,Q,R,O)=P.LT.Q.R.LT.Q.IMP.P+R.LT.O+O
 *ALT2(P,Q,R,O)=I.LT.Q.O.U.J.LT.O

ORIGINAL PAGE IS
 OF POOR QUALITY

*DLT1(P,Q,R)=P.LT.Q.R.GT.Q.IMP.P/R.LT.Q/R
 *MGE1(P,Q,R)=P.GE.Q.R.GT.Q.IMP.P*R.GE.Q*R
 *MLE1(P,Q,R)=P.LE.Q.R.GT.Q.IMP.P*R.LE.Q*R
 *MGT1(P,Q,R)=P.GT.Q.R.GT.Q.IMP.P*R.GT.Q*R
 *MLT1(P,Q,R)=P.GT.Q.R.LT.Q.IMP.P*R.LT.Q*R
 *SUBGEGT1(I,J)=I.GT.J.EQV.I-1.GE.J
 *SUBGEGT2(I,J)=I.GE.J.IMP.I.GT.J-1
 *SUBLELT1(I,J)=I.LE.J.EQV.I-1.LT.J
 *SUBLELT2(I,J)=I.LT.J.IMP.I.LE.J-1
 *SURGE1(P,Q,R)=P.GE.Q.EQV.P-R.GE.Q-R
 *SUBGT1(P,Q,R)=P.GT.Q.EQV.P-R.GT.Q-R
 *SUBGT2(P,Q,R)=P.GT.Q.IMP.P-R.GT.Q-R
 *UMLE1(P)=P.LE.Q.EQV.-P.GE.Q
 *UMLE2(P)=P.GE.Q.EQV.-P.LE.Q
 *UMLT1(P)=P.LT.Q.EQV.-P.GT.Q
 *UMLT2(P)=P.GT.Q.EQV.-P.LT.Q
 *UMSUB1(P,Q)=-(P-Q).EQ.Q-P

INEQUALITY AND LOGIC

*EQGEGTOR1(P,Q)=P.GE.Q.EQV.P.GT.Q.OR.P.EQ.Q
 *EQGEGTOR2(P,Q)=P.GT.Q.OR.P.EQ.Q.EQV.P.GE.Q
 *EQLELTOR1(P,Q)=P.LE.Q.EQV.P.LT.Q.OR.P.EQ.Q
 *GTLTNEOR1(P,Q)=P.NE.Q.EQV.P.GT.Q.OR.P.LT.Q
 *GTLTNEOR2(P,Q)=P.GT.Q.OR.P.LT.Q.EQV.P.NE.Q

INEQUALITY

*EQGE1(P,Q)=P.EQ.Q.IMP.P.GE.Q
 *EQGELE1(P,Q)=P.GE.Q.P.LE.Q.IMP.P.EQ.Q
 *GE1(P)=P.GE.P
 *GE2(P,Q,R)=P.GE.Q.Q.GE.R.IMP.P.GE.R
 *GE3(P,Q)=P.GE.Q.EQV.P-Q.GE.Q
 *GEGT1(P,Q)=P.GT.Q.IMP.P.GE.Q
 *GEGTNE1(P,Q)=P.NE.Q.P.GE.Q.IMP.P.GT.Q
 *GELE2(P,Q)=P.LE.Q.EQV.Q.GE.P
 *GELE1(P,Q)=P.GE.Q.EQV.Q.LE.P
 *GELE3(P,Q)=P-Q.LE.Q.EQV.Q-P.GE.Q
 *GTLE1(P,Q)=P.GT.Q.IMP.Q.LE.P
 *GT1(P,Q,R)=P.GT.Q.Q.GT.R.IMP.P.GT.R
 *GT2(P,Q)=P.GT.Q.EQV.P-Q.GT.Q
 *GTLT1(P,Q)=P.GT.Q.EQV.Q.LT.P
 *GTLT2(P,Q)=P.LT.Q.EQV.Q.GT.P
 *GTNE1(P,Q)=P.GT.Q.IMP.P.NE.Q
 *LE1(P,Q)=P.LE.Q.EQV.P-Q.LE.Q
 *LE3(P,Q,R)=P.LE.Q.Q.LE.R.IMP.P.LE.R
 *LE4(P)=P.LE.P
 *LELT1(P,Q,R)=P.LT.Q.Q.LE.R.IMP.P.LT.R
 *LELT2(P,Q,R)=P.LE.Q.Q.LT.R.IMP.P.LT.R
 *LELTNE1(P,Q)=P.NE.Q.P.LE.Q.IMP.P.LT.Q
 *LT1(P,Q,R)=P.LT.Q.Q.LT.R.IMP.P.LT.R
 *LT2(P,Q)=P.LT.Q.EQV.P-Q.LT.Q
 *NE1(P,Q)=P.NE.Q.IMP.Q.NE.P

LOGIC

*AELDG1(P,Q)=P.LT.Q.P.GE.Q.IMP.FALSE
 *AELDG2(P,Q)=P.GT.Q.P.Q.Q.IMP.FALSE

$AND3(A,B)=A.B.IMP.B$
 $AND4(A,B)=A.B.IMP.A.AND.B$
 $AND5(A,B)=(A.B).IMP.A$
 $AND6(A,B)=(A.B).IMP.B$
 $AND7(A)=A.AND.A.IMP.A$
 $ANDIMP1(A,B,C)=(A.IMP.B).C.IMP.(A.AND.C.IMP.B)$
 $ANDOR1(A,B,C)=(A.OR.B).C.IMP.A.AND.C.OR.E.AND.C$
 $ANDOR2(A,B,C)=A.(B.OR.C).IMP.A.AND.B.OR.A.AND.C$
 $AO1(A,B,C)=A.OR.B.C.EQV.(A.C).OR.(B.C)$
 $IMP1(A,B)=(A.IMP.B).A.IMP.B$
 $IMP2(A,B,C)=(A.IMP.B).(B.IMP.C).IMP.(A.IMP.C)$
 $IMP3(A,B)=(A.IMP.B).NOT.A.IMP.NOT.B$
 $IMO1(A,B,C)=A.CR.B.(A.IMP.C).IMP.C.OR.B$
 $*IMO2(A,B,C)=A.OR.B.(B.IMP.C).IMP.A.OR.C$
 $IMA1(A,B,C)=(A.B.IMP.C).A.IMP.(B.IMP.C)$
 $ORI(A,B)=A.IMP.A.OR.B$
 $OR2(A,B)=B.IMP.A.OR.B$
 $OR3(A,B)=A.OR.B.EQV.B.OR.A$
 $OR4(A)=A.OR.A.EQV.A$
 $ORS(A)=A.OR.NOT.A$
 $OR6(A)=A.OR.FALSE.IMP.A$
 $EQV1(A,B)=A.(A.EQV.B).IMP.B$
 $EQV2(A,B)=B.(A.EQV.B).IMP.A$
 $EQV3(A,B)=(A.EQV.B).NOT.B.IMP.NOT.A$
 $EQV4(A,B)=(A.EQV.B).NOT.A.IMP.NOT.B$
 $NOT2(P,Q)=NOT.P.EQ.Q.EQV.P.NE.Q$
 $NOT3(A,B)=NOT.A.OR.NOT.B.EQV.NOT.(A.AND.B)$
 $NOT4(A)=A.EQV.NOT.(NOT.A)$
 $NOT5(P,Q)=P.EQ.Q.EQV.NOT.(P.NE.Q)$
 $NOT6(A,B,C)=NOT.A.NOT.B.NOT.C.IMP.NOT.(A.OR.B.OR.C)$
 $NOT7(A,B)=NOT.A.NOT.B.IMP.NOT.(A.OR.B)$
 $NOTAND1(A,B)=A.AND.NOT.B.IMP.NOT.(A.AND.B)$
 $NOTORI(A,B)=NOT.A.NOT.B.IMP.NOT.(A.OR.B)$
 $FE1(A)=A.NOT.A.IMP.FALSE$

ARITHMETIC

$APT1(X,Y)=X.GE.O.Y.GE.O.IMP.R(X-Y)$
 $APT2(X)=ABS(X).GE.O$
 $APT3()=100.B23.GE.O$
 $A1(P,Q)=P+Q.EQ.Q+P$
 $A2(P,Q,R)=(P+Q)+R.EQ.P+(Q+R)$
 $A3(P)=P+Q.EQ.P$
 $*A4(P)=0+P.EQ.P$
 $*A5(P,Q,R)=P+Q.EQ.Q+R.EQV.P.EQ.Q$
 $A6(P)=P+P.EQ.2*P$
 $AM1(P,Q,R)=P*(Q+R).EQ.P*Q+P*R$
 $*AM2(P)=P.EQ.O.EQV.-P.EQ.O$
 $*AM3(P,Q)=Q.NE.O.IMP.(P+Q)/Q-1.EQ.P/Q$
 $*AMA(P,Q,R)=(P+Q)*R.EQ.P*R+Q*R$
 $AS1(P,Q)=P-Q.EQ.P+(-Q)$
 $AS2(P,Q)=-P+Q.EQ.O-P$
 $*AS3(P,Q,R)=P+Q-R.EQ.P+(Q-R)$
 $*AS4(P,Q,R)=P-Q+R.EQ.P-(Q-R)$
 $*AS5(P,Q,R)=P-Q-R.EQ.P-(Q+R)$
 $AS6(P,Q,R)=P-Q-R.EQ.P-R-Q$
 $*ASUB1(P,Q)=P+Q-Q.EQ.P$
 $D1(P)=P.NE.O.IMP.P/P.EQ.1$
 $D2(P,Q,R)=R.NE.O.IMP.(P+Q)/R.EQ.P/R+Q/R$
 $D3(P)=P.NE.O.IMP.O/P.EQ.O$
 $D4(P,Q,R)=P.EQ.Q.R.NE.O.IMP.P/R.EQ.Q/R$
 $*D5(P,Q,R)=P.CQ.Q.P.4T.N.IMP.P/R.EQ.Q/R$

EX3(P)=P**1.EQ.P
 MD1(P,Q)=Q.NE.0.IMP.P*Q/Q.EQ.P
 MU1(P,Q)=P*Q.EQ.0*P
 MU2(P,Q,R)=(P*Q)*R.EQ.P*(Q*R)
 MU3(P)=P*1.EQ.P
 MU4(P,Q,R)=R.NE.0.P*R.EQ.0*R.IMP.P.EQ.0
 *MU5(P)=P*0.EQ.0
 MU7(P,Q,R)=R.GT.0.P*R.GT.0*R.IMP.P.GT.0
 *MU6(P)=0*P.EQ.0
 MU8(P,Q,R)=R.GT.0.P*R.GE.0*R.IMP.P.GE.0
 MU9(P,Q,R)=R.GT.0.R*P.GT.0*R*Q.IMP.P.GT.0
 MU10(P,Q,R)=R.GT.0.R*P.GE.0*R*Q.IMP.P.GE.0
 MU11(P,Q,R)=R.LT.0.P*R.GT.0*R.IMP.P.LT.0
 MU12(P,Q,R)=R.LT.0.P*R.GE.0*R.IMP.P.LE.0
 MU13(P,Q,R)=P.EQ.0.IMP.P*R.EQ.0*R
 MU14(P,Q,R,D)=P.GT.0.Q.GT.0.R.GT.0.D.GT.0.IMP.P*R.GT.0*Q
 SM1(P,Q)=(-P)*Q.EQ.-P*Q
 NOT1(P,Q)=.NOT.P.NE.Q.EQV.P.EQ.Q
 SUB1(P)=P-P.EQ.0
 *SUB2(P,Q)=P-Q.EQ.0.EQV.P.EQ.0
 SUB4(P)=P-Q.EQ.P
 SUB5(P)=0-P.EQ.-P
 SUB6(P,Q)=P.EQ.Q.EQV.P-Q.EQ.0

EQUALITY

EQ1(P)=P.EQ.P
 EQ2(P,Q)=P.EQ.Q.EQV.Q.EQ.P
 EQ3(P,Q,R)=P.EQ.Q.Q.EQ.R.IMP.P.EQ.R
 EQ4(P,Q,R)=P.EQ.Q.R.EQ.Q.IMP.P.EQ.R

FLOATING POINT ARITHMETIC

FL1()=FMIN.GT.0
 FL2()=FMAX.GT.0

GREATEST COMMON DIVISOR

GCD1(I,J)=I.GT.0.J.GT.0.J-I.GT.0.IMP.GCD(I,J).EQ.GCD(I,J-I)
 GCD2(I,J)=I.GT.0.J.GT.0.IMP.GCD(I,J).EQ.GCD(J,I)
 GCD3(I)=I.GT.0.IMP.GCD(I,I).EQ.I

SUBSTITUTION

*SU1(P,Q,R,D)=P.EQ.Q.R.EQ.D.IMP.P*R.EQ.Q*Q
 *SU2(P,Q,R,D)=P.EQ.Q.R.EQ.D.IMP.P*R.EQ.Q*Q
 *SU3(P,Q,R,D)=P.EQ.Q.R.EQ.D.IMP.P-R.EQ.Q-D
 SUBST1(P,Q,R,A(R))=P.EQ.Q.A(P).IMP.A(Q)
 SUBST2(P,Q,R,A(R))=P.EQ.Q.A(Q).IMP.A(P)
 SUBST3(A,B,C,D(C))=(A.EQV.B).D(A).IMP.D(B)
 SUBST4(A,B,C,D(C))=(A.EQV.B).D(B).IMP.D(A)

SET

S1(I,J)=I.GT.0.FOV.(I,J).P*Q(I)

```

S4(I,J,S,L,A(L))=I.INSIDE.S.EQV.A(I).INSIDE.SET(A(J),J.IN.S)
S5(I,J,K,S,L,N(L))=E(I(K.IN.S,I.EQ.N(K)).EQV.I.INSIDE.SET(N(J),J.IN.S)
S6(I,J,K,S,L,N(L))=I.INSIDE.SET(N(J),J.IN.S).EQV.EI(K.IN.S,I.EQ.N(K))
S7(I,J,K,S,T,L,A(L))=E(I(I.IN.S,A(I)).(A(J).IMP.J.INSIDE.T).IMP.EI(K.IN.T,A(K))
S8(I,J,L,A(L))=A(J).IMP.(A(I).I.IN.(J..J))
S9(I,J,K,L,A(L),M,B(M))=(A(I).I.IN.S).(A(J).IMP.B(J)).IMP.(B(K),K.IN.S)
S10(I,J,K,L,A(L))=J.GT.K.IMP.FI(I,J.LE.I.AND.I.LE.K.IMP.A(I))
S10A(I,R,A(R))=FA(I.IN.(I),A(I))
S11(I,J,K)=I.GE.J.I.LE.K.IMP.I.INSIDE.(J..K)
S12(I,J,S,T,M,N,A(M,N))=(A(J,I).I.IN.S.J.IN.T).EQV.((A(J,I).I.IN.S).J.IN.T)
S12A(I,J,S,T,M,N,A(M,N))=(A(I,J).I.IN.S.J.IN.T).EQV.((A(I,J).I.IN.S).J.IN.T)
S13(I,J,S,T,M,N,A(M,N))=(A(J,I).I.IN.S.J.IN.T).EQV.(A(J,I).J.IN.T.I.IN.S)
S15(I,J,L,A(L))=(A(I).I.IN.(J..J)).EQ.A(J)
S16(I,J,L,A(L))=A(J).IMP.FI(I,J.LE.I.AND.I.LE.J.IMP.A(I))
S17(I,S,M,A(M))=A(I).IMP.(A(I).I.IN.S)

```

IF EXPRESSION

```

IF1(I,J,K,A)=I=(.IF.A.THEN.J.ELSE.K).A.IMP.I=J
IF2(I,J,K,A)=I=(.IF.A.THEN.J.ELSE.K).NOT.A.IMP.I=K
IF3(I,J,K,A)=I=J.A.IMP.I=(.IF.A.THEN.J.ELSE.K)
IF4(I,J,K,A)=I=K.NOT.A.IMP.I=(.IF.A.THEN.J.ELSE.K)
IF5(I,J,K,R,A(R))=.IF.I=J.THEN.A(I).ELSE.K.EQ..IF.I=J.THEN.A(J).ELSE.K
IF6(I,J,K,M,B(M),P(M),R(M),Q(M))=(P(J)=Q(J)).J.IN.S.IMP.((.IF.B(I).THEN.P(I).ELSE.R(I))=Q(I).I.IN.S).NOT.(B(K),K.IN.S)
IF7(I,J,K,M,B(M),P(M),R(M),Q(M))=(P(J)=Q(J)).J.IN.S.IMP.((.IF.B(I).THEN.P(I).ELSE.R(I))=Q(I).I.IN.S).NOT.(B(K),K.IN.S)
IF8(K,J,I,M,B(M),P(M),R(M),Q(M))=((.IF.B(K).THEN.P(K).ELSE.R(K))=Q(K),K.IN.S).EQV.((.IF.B(I).THEN.P(I).ELSE.R(I))=Q(I).I.IN.S).(B(J),J.IN.S))
IF9(A,B,P,Q,R,S)=P.EQ.Q.NOT.A..NOT.B.IMP.P=.IF.A.THEN.R.ELSE..IF.B.THEN.S.ELSE.Q

```

UNION AND INTERSECTION

```

UNION1(I,J,K,I1,I2,J1,J2,L,A(L))=FI(I,I1.LE.I.AND.I.LE.I2.IMP.A(I)).FI(J,J1.LE.J.AND.J.LE.J2.IMP.A(J)).I2.EQ.J1-1.IMP.FI(K,I1.LE.K.A
ND.K.LE.J2.IMP.A(K))
UNION2(I,J,K)=I.LE.J.J.LE.K.IMP.(I..J-1).UNION.(J..K).EQ.(I..K)
UNION3(I,J,K)=I.LE.J.J.LE.K.IMP.(J..K).UNION.(I..J-1).EQ.(I..K)
INST1(I,J,K,L)=K.LE.L.J.LT.K.IMP.(I..J).INST.(K..L).EQ.(I)

```

SUMMATION

```

SUM1(I,J,K,L,S,T,M,P(M))=SUM(P(I).I.IN.S)+SUM(P(J).J.IN.T).EQ.SUM(P(K),K.IN.S.UNION.T)-SUM(P(L),L.IN.S.INST.T)
SUM2(I,J,K,L,M,N,A(N))=L.LE.M.K.EQ.L-1.IMP.SUM(A(I).J.LE.I.AND.I.LE.K)+SUM(A(I).L.LE.I.AND.I.LE.M).EQ.SUM(A(I).J.LE.I.AND.I.LE.M)
SUM3(I,J,K,L,P(L))=J.GT.K.IMP.SUM(P(I).J.LE.I.AND.I.LE.K).EQ.Q
SUM4(I,J,K,L,P(L))=J.EQ.K.IMP.SUM(P(I).J.LE.I.AND.I.LE.K).EQ.P(J)

```

MISCEL.

```

ABSLTLE1(P,Q,R,S)=ABS(P).LT.Q.ABS(R).LT.S.IMP.ABS(P)*ABS(R).LT.Q*S
ABSS(P,Q)=ABS(P*Q)=ABS(P)*ABS(Q)
ABS6(P,Q)=P.EQ.Q.IMP.ABS(P).EQ.ABS(Q)
ABS7(P,Q)=P.LT.Q.P.GE.Q.IMP.ABS(P).LT.ABS(Q)
ABS8(P,Q)=P.LT.Q.Q.GE.1.IMP.P/Q.LT.Q
ABS9(P)=P.GE.Q.IMP.ABS(P).EQ.P
ABS10(P,Q)=Q.NE.C.IMP.ABS(P/Q).EQ.ABS(P)/ABS(Q)
ABS11(P,Q)=Q.LE.P.Q.GE.Q.IMP.Q.LE.ABS(P)
ABS12(P,Q)=ABS(P).LT.1.AND.Q.GE.1.IMP.ABS(P)/Q.LT.1.BQ
ABS13(P,Q)=ABS(P*Q)-ABS(P)*ABS(Q)

```

ORIGINAL PAGE IS
OF POOR QUALITY

ABS16(P,Q)=P.GE.O.P.LT.Q.IMP.ABS(P).LT.ABS(Q)
 ABS17(P,Q)=P.GE.-Q.AND.P.LE.O.Q.GE.O.IMP.ABS(P).LE.Q
 ABS18(P)=ABS(-P).EQ.ABS(P)
 ABS19(P,Q)=P.EQ.O.Q.GE.O.IMP.ABS(P).EQ.Q
 ABS20(P,Q,R)=ABS(Q).GT.R.R.GT.O.IMP.ABS(P/Q)=ABS(P)/ABS(Q)
 AEQGE2(P,Q,R,S)=P.EQ.O.Q.R.GE.S.IMP.P+R.GE.O+S
 AEQLE1(P,Q,R,S)=P.LE.O.Q.R.EQ.S.IMP.P+R.LE.O+S
 ALE1(P,Q,R,S)=P.LE.O.Q.R.LE.S.IMP.P+R.LE.O+S
 ALTLE1(P,Q,R,S)=P.LT.O.Q.R.LE.S.IMP.P+R.LT.O+S
 ANDNOT1(A,B)=A.AND.H.IMP..NOT.(A.AND..NOT.B)
 AND8(A,B)=A.AND.B.IMP.A
 AND9(A,B)=A.AND.B.IMP.B
 AQ1(P,Q,R,S)=P.EQ.(Q.R).ABS(Q)*S.LT.1.B0.IMP.DIR(P*S)
 AQ2(P,Q,R,S,T)=P.EQ.(Q.R)*S.DIR(P).T.EQ.AR(P).IMP.T.EQ.Q*S
 AQ3(X,Y,Z,W,S)=X=(Y.Z).ABS(Y)*W.LT.S.S.LT.1.B0.IMP.ABS(X).LT.Z
 AQ4(X,Y,Z)=X=AR(Y).ABS(Y).LT.Z.IMP.ABS(X)/ABS(Z)
 DEQ1(P,Q,R)=P.EQ.Q/R.R.NE.O.IMP.ABS(P).EQ.ABS(Q)/ABS(R)
 DIV2(P,Q,R,S)=P.LE.O.Q.P.EQ.S.S.NE.O.IMP.P/R.LE.Q/S
 DIV3(P,Q,R,S)=P.LE.O.Q.R.LE.S.P.GT.O.IMP.R/Q.LE.S/P
 DIV4(P,Q)=P.GE.O.Q.GT.O.IMP.P/Q.GT.O
 DIV6(P,Q,R,S)=P.LT.Q.R.EQ.S.S.NE.O.IMP.P/R.LT.Q/S
 DLE1(P,Q,R)=P.LE.Q.R.NE.O.IMP.P/R.LE.Q/R
 DLTGT1(P,Q,R)=P.LT.Q.R.GT.O.IMP.P*R.LT.Q*R
 EQORLT1(P,Q,R)=P.EQ.O.Q.R.P.EQ.R.Q.LT.R.IMP.ABS(P).LE.R
 EQVNOTAND1(A,B,C,D)=(A.EQV.C.AND.O).(B.EQV.C.AND..NOT.D).IMP.(B.EQV..NOT.A)
 GEGTNE1(P,Q,R)=P.GE.O.Q.GT.R.IMP.P.NE.R
 GEGT2(P,Q,R)=P.GE.O.Q.GT.R.IMP.P.GT.R
 IR5(P,Q)=P.GE.O.Q.GE.C.IMP.R(P-Q)
 IR1(I)=R(I).IMP.R(I.ABS(I))
 I=2(I,J)=R(I).R(J).I.GE.O.J.GE.O.IMP.N(I-J).EQ.I-J
 IR3(I,J)=I.EQ.J.R(J).IMP.R(I)
 IR4(I,J)=I.EQ.-J.R(J).IMP.R(I)
 IR6(P)=ABS(P).LT.1.B0.IMP.R(P)
 LEGE1(P,Q,R,S)=P.LE.Q.R.LE.S.P.GE.O.Q.R.GE.O.IMP.P*R.LE.Q*S
 LELT4(P,Q,R,S)=P/O.LT.R.S.LE.P.IMP.S/Q.LT.R
 LELT5(P,Q,R,S)=P/Q.LT.R.S.GT.O.Q.NE.O.IMP.P/S.LT.R
 LE5(X,Y,Z,W)=X.LE.Y.Z.LE.W.IMP.X+Z.LE.Y+W
 LTGE1(P,Q,R,S)=P.LT.Q.R.LE.S.P.GE.O.Q.R.GE.O.IMP.P*R.LT.Q*S
 LTGE2(P,Q,R,S)=P.LE.Q.R.LT.S.P.GE.O.Q.R.GE.O.IMP.P*R.LT.Q*S
 LTLE1(P,Q,R)=P.LT.Q.Q.LE.R.IMP.P.LT.R
 MLT2(P,Q,R)=P.LT.Q.R.GT.O.IMP.P*R.LT.Q*R
 MULTEQ1(P,Q,R)=P.EQ.MULT(Q,R).IMP.P=Q*R
 MULTEQ2(P,Q,R)=P.EQ.Q*R.IMP.ABS(P).EQ.ABS(Q)*ABS(R)
 MULT1(P,Q)=MULT(P,Q).EQ.P*Q
 MULT2(P,Q,R,S)=P.LT.Q.Q*R.LT.S.IMP.P*R.LT.S
 MULT3(P,Q)=MULT(ABS(P).ABS(Q)).EQ.ABS(MULT(P,Q))
 MULT4(P,Q,R)=ABS(P).LT.R.ABS(Q).LT.1.B0.IMP.ABS(MULT(P,Q)).LT.R
 MULT5(P,Q,R)=ABS(Q).LT.R.ABS(P).LT.1.B0.IMP.ABS(MULT(P,Q)).LT.R
 MU15(P,Q,R)=P.GT.O.P.LT.Q.IMP.P*R.LT.Q*R
 NEG1(P,Q)=P.GT.Q.Q.NE.O.IMP.O.NE.O
 NELE1(P,Q)=P.LE.Q.P.NE.O.IMP.O.NE.O
 NF3(P,Q,R,S)=ABS(P).LT.R.ABS(Q).LE.S.IMP.ABS(N(P-Q)).LT.R+S
 NF4(P,Q)=ABS(P).LT.O.SB0.ABS(Q).LT.O.SB0.IMP.ABS(N(P-Q)).LE.ABS(P)+ABS(Q)
 NF5(P,Q)=ABS(P).LT.C.SB0.ABS(Q).LT.O.SB0.IMP.ABS(N(P+Q)).LE.ABS(P)+ABS(Q)
 NF6(P,Q)=ABS(P).LT.7.SB0.ABS(Q).LT.O.SB0.IMP.N(P+Q)=P+Q
 NF7(P,Q)=P.EQ.O.ABS(Q).LT.1.B0.IMP.ABS(N(P-Q))=ABS(Q)
 NOTAND1(A,B)=0.IMP..NOT.(A.AND..NOT.B)
 NOTEDGT1(P,Q)=P.EQ.Q.IMP..NOT.(P.GT.Q)
 NOTEQ4(P,Q)=.NOT.P-Q.EQ.O.IMP..NOT.P.EQ.Q
 NOTGEGT1(P,Q)=.NOT.(P.EQ.Q).P.GE.Q.IMP.F.GT.Q
 NOTGE1(P,Q)=.NOT.(P.GE.Q).EQV.P.LT.Q
 NOTGE2(P,Q)=P.GE.Q.EQV..NOT.P.LT.Q
 NOTGE3(P,Q)=.NOT.P.GE.O.EQV..NOT.P.GT.Q
 NOTLTGT1(P,Q)=.NOT.P.LT.Q..QV.P.GE.Q

ORIGINAL PAGE IS
 OF POOR QUALITY

NOTORAND1(A,B)=NOT.(A.OR.B).EQV..NOT.A.AND..NOT.B
 NOTORAND2(A,B)=NOT.B.IMP..NOT.(A.AND.B)
 NOTORAND3(A,B)=NOT.A.IMP..NOT.(A.AND.B)
 NOTORAND5(A,B)=A.OR.B..NOT.A.IMP.B
 NOTORAND6(A,B,C)=A.OR.B.AND.C.IMP..NOT.(NOT.A.AND.B.AND..NOT.C)
 NOTORAND7(A,B,C,D,E,F)=(A.EQV.B.AND.C.OR.D.AND..NOT.E.AND.F)..NOT.F.IMP.(A.EQV.B.AND.C)
 NOTORAND8(A,B,C,D,E,F)=A.OR.B.AND.C.(B.EQV.D.AND.E).(D.EQV..NOT.A.AND.F).IMP..NOT.(D.AND..NOT.E)
 NOT8(A,B,C)=A.CR..NOT.B.OR..NOT.C.IMP..NCT.(NOT.A.AND.B.AND.C)
 NOTORAND9(A,B,C)=A.AND.(B.OR.C).IMP..NOT.(A.AND..NOT.B.AND..NOT.C)
 N1(P)=N(P).LE.1.B0
 N2(P)=ABS(N(P)).LT.1.B0
 N3(P)=N(P).IMP.ABS(N(P))
 SRLMT1(I,J)=IABS(I).LT.J.IMP.I.EQ.SRLMT(I,J)
 SRLMT2(I,J)=IABS(I).GE.J.I.LT.0.IMP.-J.EQ.SRLMT(I,J)
 SRLMT3(I,J)=IABS(I).GE.J.I.GE.0.IMP.J.EQ.SRLMT(I,J)
 SRLMT5(P,Q)=ABS(SRLMT(P,Q)).LE.ABS(Q)
 SRLMT6(P,Q)=ABS(P).LT.ABS(Q).IMP.ABS(SRLMT(P,Q))=ABS(P)
 SUBST5(A,B,C,D(C))=(A.IMP.B).D(A).IMP.D(E)
 IABS1(I,J)=I.EQ.IABS(J).IMP.I.GE.0
 *BS1(S.I.J,K,A(K))=FA(I.IN.S.A(I)).OR.EI(J.IN.S..NOT.A(J))
 EIFA1(S.I.J,K,R,A(R))=FA(I.IN.S.A(I)).EQV..NOT.EI(J.IN.S..NOT.A(J))
 EI1(I,J,K,L,P(L),M,Q(M))=EI(I.J.LE.I.AND.I.LE.K.AND.P(I).EQ.Q(I)).P(J).NE.Q(J).IMP.EI(I.J+1.LE.I.AND.I.LE.K.AND.P(I).EQ.Q(I))
 EI2(I,J,K,L,P(L),M,Q(M))=EI(I.J.LE.I.AND.I.LE.K.AND.P(I).EQ.Q(I)).J.GT.K.IMP..FALSE.
 EI3(I,J,K,L,M,A(M))=A(J).K.LE.J.J.LE.L.IMP.EI(I,K.LE.I.AND.I.LE.L.AND.A(I))
 EI4(I,J,M,L)=MOD(I,J).EQ.M.IMP.FI(L.IN.(C..I)).I.EQ.J*L+M
 *LEMMA1(I,K,L)=(K.GE.0.K.LE.(J-1)*4).K.EC.4*L.IMP.L.INSIDE.(0..J-1)
 MIS1(J,P,S,I,N(I),M,A(M))=J.GE.0.(A(N(I))=P.I.IN.S).IMP.(A(N(I))={.IF.N(I)=N(J).THEN.P.ELSE.P}).I.IN.S.UNION.(J..J))
 MISS(I,L1,L2,J,P,L,N(L),M,A(M))=L2.EQ.J-1.FI(I.L1.LE.I.AND.I.LE.L2.IMP.A(N(I))=P).IMP.FI(I.L1.LE.I.AND.I.LE.J.IMP.(.IF.N(I)=N(J).THE
 N.P.ELSE.A(N(I))=P)
 *SET1(K,L,S,M,N,A(M,N))=(A(K,L).K.IN.(I).L.IN.S)
 *SET2(I,J,K,L,M,N,A(M,N))=A(J,L).IMP.(A(I,K).I.IN.(J..J).K.IN.(L..L))

THEOREMS NOT YET JUSTIFIED

PR1(I,K,L)=(I.NE.K*L.K.IN.(2..I-1).L.IN.(2..I-1)).EQV.PRIME(I)
 PR2(I)=I.GT.0.I.LT.4.IMP.PRIME(I)
 PR3(I,J)=MOD(I,J).EQ.0.J.GE.2.I.GT.0.J*J.LE.I.IMP..NOT.PRIME(I)
 PR4(I,J,K,L)=MOD(I,J).NE.0.(I.NE.K*L.K.IN.(2..J-1).L.IN.(2..I-1)).I.GT.0.J.GE.2.(J+1)*(J+1).LE.I.IMP.(I.NE.K*L.K.IN.(2..J).L.IN.(2..
 I-1))
 PR5(I,J,K,L)=MOD(I,J).NE.0.I.GT.0.J.GE.2.(I.NE.K*L.K.IN.(2..J-1).L.IN.(2..I-1)).(J+1)*(J+1).GT.I.IMP.PRIME(I)
 BS2()=((FA(I.IN.(0..(N-1)).X.NE.A(4*I))).OR.(EI(J.IN.(R3..R5).X.EQ.A(4*J))).X.GT.A((R3+R5)/2**1*4)).IMP.((FA(I.IN.(0..(N-1)).X.
 NE.A(4*I))).OR.(EI(J.IN.(((R3+R5)/2**1+1)..R5).X.EQ.A(4*J))))
 BS3()=((FA(I.IN.(0..(N-1)).X.NE.A(4*I))).OR.(EI(J.IN.(R3..R5).X.EQ.A(4*J))).X.LE.A((R3+R5)/2**1*4)).IMP.((FA(I.IN.(0..(N-1)).X.
 NE.A(4*I))).OR.(EI(J.IN.(R3..(R3+R5)/2**1).X.EQ.A(4*J))))
 SB4()=((FA(I.IN.(0..(N-1)).X.NE.A(4*I))).OR.(EI(J.IN.(R3..R5).X.EQ.A(4*J))).(R3.EQ.R5).(X.NE.A(R3*4)).IMP.(FA(I.IN.(0..(N-1)).X.
 NE.A(4*I)))

ORIGINAL PAGE IS
 OF POOR QUALITY

--- USER DEFINED MACROS ---

```

CCRUZFD=CFDIN1.OR.(CFD2.AND.LOCFD03=0)
CFDIN1=(TMEFLG.NE.0).AND.T1CRUZFD
T1CRUZFD=(AND(PCUMWD.T12)=0).OR.(ILSON=0).OR.(GRUND.NE.0)
CFD1=CFDIN2
CFDIN2=(TMEFLG.NE.0).AND.(.NOT.T1CRUZFD)
CFDIN3=CFDIN2.AND.START=0
CFD3=CFDIN2.AND.TFD3
TFD3=(ABS(YETA01)-LIMLOC.LT.0).OR.N(ABS(Psie01)-DEG2).LT.0
CFD4=CFDIN2.AND.(TFD3.OR.(GUID2D=0))
CFD42=CFD4.AND.LBS.NE.0.AND.LOCFD03=0
CFD2=(CFD4.AND.(LBS=0)).OR.(CFDIN2.AND.(.NOT.TFD3).AND.(GUID2D.NE.0))
CLANDFD=.NOT.CFDIN1.AND.CFD2.AND.LOCFD03.NE.0.OR.CFD4.AND.LBS.NE.0
CFD52=CLANDFD.AND.(HRAD.LT.K1000)
CFD6=CLANDFD
CFD7=CLANDFD.AND.TFD7
CFD62=CLANDFD.AND.(.NOT.TFD7)
TFD7=YETA01*PSIEC1.GT.0.OR.LOCSTR01*PSIE01.LT.0.OR.N(ABS(Psie01)-DEG20).LT.0
CFD8=CFD7.AND.(HRAD.LT.A1200)
CFD9=CCRUZFD.AND.(GUID2D.NE.0)
CFD10=CFD9.OR.(CFD7.AND.(HRAD.GE.A1200))
CFD11=CFD62.OR.CFD8.OR.CFD9.OR.CFD10
CFD112=.NOT.CEXIT1
CFD13=CFD11.AND.(LOCFD06=0)
CFD111=CFD11.AND.(LOCFD06.NE.0)
CFD14=(CFD111.AND.TFD14).OR.(CFD13.AND.(GUID2D=0))
TFD14=(LCCVLD=0).OR.(MPILS01=0)
CFD12=(CFD111.AND.(.NOT.TFD14)).OR.(CFD13.AND.(GUID2D.NE.0))
CVCRUZ=CFD112.AND.TVCRUZ
TVCRUZ=LOCFD06=0.OR.(GSFD02=0.AND.GSBSNS=0)
CVLAND1=CFD11.AND.TVLAND1
TVLAND1=LOCFD06.NE.0.AND.GSFD02=0.AND.GSBSNS.NE.0
CFD15=CFD11.AND.(.NOT.TVCRUZ)
CFD152=CFD15.AND.(FLARE02=0)
CFD155=CFD15.AND.TFD155
TFD155=N(SRLMT(KFLR01.MSKL15)*2**8*TP403-K4).LT.0
TFD156=N(N(GAMMAC01+GSA)+GSA).LT.0
CVCRUZ2=CVCRUZ.AND.(GUID3D.NE.0)
CVCRUZ3=CVCRUZ.AND.TVCRUZ3
TVCRUZ3=(GUID3D.NE.0).AND.(MDA.GE.HRAD).AND.(TIMER.NE.BINTME)
CEXIT0=CVCRUZ2.AND.TEXIT0
CEXIT1=TMEFLG=0.OR.CCRUZFD.AND.GUID2D=0
CEXIT2=(CFD15.AND.(GSVLD=0)).OR.(CVCRUZ.AND.TEXIT2)
CEXIT3=(CFD15.AND.(GSVLD.NE.0)).OR.(CVCRUZ.AND.TEXIT3)
TEXIT2=(GUID3D=0).OR.(TVCRUZ3.AND.(PCMVLD.NE.0))
TEXIT3=(GUID3D.NE.0).AND.(MDA.LT.HRAD).OR.((TIMER.NE.BINTME).AND.(PCMVLD=0))
TEXIT0=(GUID3D.NE.0).AND.(MDA.GE.HRAD).AND.(TIMER=BINTME)
1111111111
MPILS01=AND(PCUMWD.T12)
STARTC1=0
LOCFD01=0
GSFD01=0
START02=SET
TP101=0
TP201=0
TP301=0
TP401=0
TP501=0
TP601=0
TP701=0

```

```

FLARE01=0
TP102=.IF.CFDIN3.THEN.0.ELSE.TP1
TP202=.IF.CFDIN3.THEN.0.ELSE.TP2
TP302=.IF.CFDIN3.THEN.0.ELSE.TP3
TP402=.IF.CFDIN3.THEN.0.ELSE.TP4
TP502=.IF.CFDIN3.THEN.0.ELSE.TP5
TP602=.IF.CFDIN3.THEN.0.ELSE.TP6
TP702=.IF.CFDIN3.THEN.0.ELSE.TP7
LOCFO03=.IF.CFDIN3.THEN.0.ELSE.LOCFO
LOCOC02=.IF.CFDIN3.THEN.0.ELSE.LOCOC
FLARE02=.IF.CFDIN3.THEN.0.ELSE.FLARE
START03=.IF.CFDIN3.THEN.SET.ELSE.START
YETA01=MULT(LOCDEV,K1)
PSIE01=N(TK-RWYHDG)
LOCOC03=SET
LOCOC04=.IF.CFO3.THEN.SET.ELSE.LOCOC02
LOCFO04=SET
RLFD01=ROLL
LOCFO05=.IF.CFO42.THEN.SET.ELSE.LOCFO03
RLFD02=.IF.CFO42.THEN.ROLL.ELSE.RLFD
TP103A=N(N(YETA01/PERIOD/PERIOD*K5-TP102)/TAU1*DELTAT/2**3+TP102)
TP103B=N(N(SRLMT(YETA01,LIMY)/PERIOD/PERIOD*K5-TP102)/TAU1*DELTAT/2**3+TP102)
TP103=.IF.LOCOC04=0.THEN.TP103A.ELSE.TP103B
TP203=GS/PERIOD*K6*PSIE01*KDAMP*2**2
TP204=N(TP203-TP702)
TP703=N(TP204*DELTAT/TAU2/2**6+TP702)
TP205=.IF.CFO52.THEN.N(TP203-TP702).ELSE.TP203
TP704=.IF.CFO52.THEN.TP703.ELSE.TP702
LOCSTR01=-(SRLMT(N(TP205-TP103),LIM180)*2**3)
TP104=.IF.CCRUF02.THEN.TP102.ELSE.TP103
TP206=.IF.CCRUF02.THEN.TP202.ELSE.TP205
TP705=.IF.CCRUF02.THEN.TP702.ELSE.TP704
CAPT01=0
CAPT02=SET
AC01=.IF.CFD9.THEN.LATSTR.ELSE.LOCSTR01
AC02=.IF.CFD62.THEN.A0.ELSE..IF.CFO3.THEN.SRLMT(LOCSTR01,DEG15).ELSE.SRLMT(AC01,DEG25)
RCMVL001=0
PCMVLD01=0
LOCFO06=.IF.CFDIN1.THEN.0.ELSE..IF.CLANDFD.THEN.LOCFO05.ELSE.LOCFO03
RLFD04=N(SRLMT(N(AC02-RLFD02)/2**2/TAU3,DEG4)*DELTAT+RLFD02)
RCMD01=SRLMT(N(RLFD04-ROLL),DEG25)
RCMVL002=SET
RCMVLD03=0
LGAMC01=GAMMA
TP303=N(N(GSDEV*K1*2**8-TP302)/TAU4*DELTAT/2**4+TP302)
TP403=N(N(TP303/C180BF*K3+GSA)-K4)
HREF01=.IF.HOCF*K2.LT.C.THEN.(HOCF*K7)*TF/K8.ELSE.(K10+K7)*TF/K8
HREF02=.IF.CFD152.THEN.HREF01.ELSE.HREF
FLARE03=.IF.KFLR01-MSKL15.LT.0.THEN.SET.ELSE.0
KFLR01=.IF.CFD15.THEN.HRAD/HREF02.ELSE.KFLR
GAMMAC01=.IF.TFD155.THEN.N(SRLMT(KFLR01,PSKL15)*2**8*TP403-K4).ELSE.GAMMAC
LGAMC02=.IF.CVLAND1.THEN.GAMMA.ELSE.LGAMC
TLGAMC0=N(SRLMT(N(A0-LGAMC02)/TAU5/2**3,LIMPR)*DELTAT+LGAMC02)
TLGAMC1=N(SRLMT(N(-N(N(GAMMAC01+GSA)+GSA)-GAMMAC01)-LGAMC02)/TAU5/2**3,LIMPR)*DELTAT+LGAMC02)
LGAMC03=.IF.TFD155.THEN.LGAMC04.ELSE.TLGAMC0
LGAMC04=.IF.TFD156.THEN.TLGAMC1.ELSE.TLGAMC2
TLGAMC2=N(SRLMT(N(GAMMAC01-LGAMC02)/TAU5/2**3,LIMPR)*DELTAT+LGAMC02)
GAMCMD01=N(LGAMC03-GAMMA)
TP603=N(PITCH-TP502)
TP503=N(TP603*DELTAT/TAU6/2**3+TP502)
PCMD01=SRLMT(N(-(TP603*FDBKG*2**3)+GAMCMD01),DEG6)
RCMVLD04=.IF.CFD12.THEN.SET.ELSE.0
PCMVLD02=RCMVLD04
PCMD02=N(SRLMT(SRLMT(VERSTR,C5H15)/VG5/2**2/KHD/PIB2,DEG6)-PCMD)/C3B2*DELTAT/2**2+PCMD)

```

ORIGINAL PAGE IS
OF POOR QUALITY

GSFD03=.IF.CVLAND1.THEN.SET.ELSE.GSFD02
 PCMD03=.IF.CVCRUZ2.THEN.PCMD02.ELSE.PCMD04
 PCMD04=.IF.TVCRUZ.THEN.PCMD.ELSE.PCMD01
 START04=.IF.CFDIN1.THEN.0.ELSE.START03
 GSFD02=.IF.CFDIN1.THEN.0.ELSE.GSFD
 YETA02=.IF.CFDIN1.THEN.YETA.ELSE.YETA01
 PSIE02=.IF.CFDIN1.THEN.PSIE.ELSE.PSIE01
 CAPT03=.IF.CFD7.THEN.SET.ELSE.0
 RCMVLD04=.IF.CFD12.THEN.SET.ELSE.0
 LOCSTR02=.IF.CCRUZFD.THEN.LOCSTR.ELSE.LOCSTR01
 CAPT04=.IF.CCRUZFD.THEN.CAPT.ELSE.CAPT03
 2222222222
 SPEC=A0=0.A1200=1200.B23.C180BF=180.B15.C382=3.B2.C5815=5.B15.DEG2=0.0111111180.DEG4=0.0222222280.DEG6=0.0333333380.DEG15=0.08333333
 B0.DEG20=0.1111111180.DEG25=0.1388888880.DELTAT=0.0580.OR.DELTAT=0.180.FOBKG=0.883.ABS(GAMMA).LT.1.B0.ABS(GAMMAC).LT.1.B0.ABS(GAMCMD
).LT.1.B0.ABS(GS).LE.0.625815.ABS(GSA).LE.0.0277777780.ABS(GSDEV).LE.0.0011111186.HRAD.GE.0.AND.HRAD.LT.1500.B23.HREF.GE.50.B15.AND.
 HREF.LT.1.B0.ABS(HDCF).LE.1.B8.ABS(KFLR).LE.1.B15.KDAMP=0.980.ABS(KHD).LT.1.B0.AND.ABS(KHD).GT.0.01680.ABS(KFLR).LE.1.B15.K1=700.B11
 .K2=3.B15.K3=0.0380.K4=0.K5=0.3980.K6=0.65380.K7=2.B15.K8=-1.685.K10=-3.B15.K1000=837.B23.K1200=926.B23.ABS(LATSTR).LT.1.30.LGAMC.GE
 .-0.0555555580.AND.LGAMC.LE.0.LIMY=400.B23.LIMLOC=100.B23.LIMPR=0.0033333380.LIM180=003777777.ABS(LOCDEV).LE.0.0111111185.ABS(LOCSTR
).LT.1.B0.MDA=250.B23.ABS(PCMD).LE.0.0333333380.PERIOD=35.B10.PIB2=3.141592682.ABS(PITCH).LT.1.B0.ABS(PSIE).LT.1.B0.ABS(RCMD).LE.0.1
 388888880.ABS(ROLL).LT.1.B0.ABS(RWYHDG).LT.1.B0.RESET=000000000.SET=011111111.TAU1=3.83.TAU2=60.B6.TAU3=1.B2.TAU4=3.84.TAU5=1.B3.TAU
 6=1.583.TF=16.B5.TIMER.GE.0.AND.TIMER.LT.1.B0.ABS(TK).LT.1.B0.ABS(TP1).LE.1.B3.ABS(TP2).LE.1.B3.ABS(TP3).LE.1.B13.ABS(TP4).LT.1.B0.A
 BS(TP5).LT.1.B0.ABS(TP6).LT.1.B0.ABS(TP7).LE.1.B3.T12=0400.ABS(VERSTR).LE.1.B8.VGS.GE.100.B15.ABS(YETA).LT.1.B0.ABS(RCMD).LE.0.13888
 88880.BINTME.GE.0.AND.BINTME.LT.1.B0.MSKL15=000077777.ABS(RLFD).LT.1.B0.ICA

ORIGINAL PAGE IS
 OF POOR QUALITY

```

1  ESTLIMIT = 1000000
2  EANCHOR = 1
3  STRIM = 1
4  DEFINE (' INTERM(X)Z,Y,L,H,DP,X0,HH,XH,COMMAS')
5  DEFINE (' INFX(3T)H,E,SH,IPRE,SHM,AYN,SUB,QJEXP,N,INNO,ALL,L')
6  DEFINE ('NOPARA(TXT)X1')
7  DEFINE ('WANG(ANTE,CONSEQ)PHI,PSI')
8  DEFINE ('SFPE(TEXT)X1,X2,X3')
9  UNOP = 'NOT'
10 BINOP = 'AND' | 'IMP' | 'OR' | 'EQV'
11 FORMULA = ' ' UNOP . OP ((' BAL . PHI ')) | ' ' BINOP . OP (('
+ BAL . PHI ' ' GAL . PSI '))
12 ATCM = (' ' BAL . ' ' ) . A
13 SYSTN = (' $SUM' | '$MIN' | '$MAX' | '$PROD' | '$ARRAY' | '$SET')
14 LOP = (' $NOT' | '$AND' | '$OR' | '$IMP' | '$EQV')
15 FUNC = 'SUM' | 'PROD' | 'MIN' | 'MAX'
16 BADD = '*' | '/' | '+' | '-'
17 UADP = ANY('+-')
18 MUOP = ANY('*/')
19 HD = ('H' | 'L' | 'A' | 'C')
20 ALPHA = '%AUCDEFGHIJKLMNOPQRSTUVWXYZ'
21 NUM = '123456789'
22 AN = ALPHA NUM
23 IO = ANY(ALPHA) (SPAN(AN) | NULL)
24 VAR = ID ((' BAL ')) | NULL
25 ITRC = (SPAN(NUM) | NULL)
26 ITR = ANY(NUM) ITRC
27 SCLNUM = (ITR . ' ' | ITRC . ' ' ITR) . B ITR
28 DIGIT = '01234567'
29 OCTNUM = '0' SPAN('1234567')
30 INTEGER = SCLNUM | ITR | OCTNUM
31 DELIM = ' ' | '.' | ',' | '(' | ')' | POS(0)
32 DULIMP = ' ' | '.' | ',' | '(' | ')' | RPOS(0)
33 BALUCU = BAL (' ' | '.' | ',' | '(' | ')' | RPOS(0))
34 PELTN = ('GT' | 'GE' | 'EQ' | 'NE' | 'LT' | 'LE') . RENA
35 VIC = TABLE(40)
36 SS1 = TABLE(44)
37 SS2 = TABLE(44)
38 OPT = TABLE(3)
39 OT = TABLE(22)
40 XT = TABLE(40)
41 IPH = TABLE(30)
42 STK = ARRAY('30')
43 STP = ARRAY('50')
44 CUP = ARRAY('51')
45 Z = ALPHA
46 AA1 Z LFN(1) . Y = :F(AA2)
47 VIC(Y) = 1
48 SS1(Y) = 'PRL' : (A1)
49 AA2 Z = NUM
50 AA3 Z LFN(1) . Y = :F(AA4)
51 VIC(Y) = 2
52 SS1(Y) = 'PRD' : (AA3)
53 AA4
54 VIC(Y) = 3
55 VIC(Y) = 4

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

56 OPT<'**'> = 1
57 OPT<'*'> = 1
58 OPT<'+'> = 1
59 OPT<'-'> = 1
60 OPT<'/'> = 1
61 SS1<'-'> = 'PRJM' ; SS1<'+'> = 'PRUP'
63 SS1<'('> = 'PRLP' ; SS1<'>'> = 'PRNULL'
65 SS1<'-'> = 'PRNOT' ; SS1<'.'> = 'PRN'
67 SS2<'-'> = 'PRNOT1'
68 SS2<'('> = 'PRP' ; SS2<'+'> = 'PRDD'
70 SS2<'-'> = 'PRDD' ; SS2<'*'> = 'PRT'
72 SS2<'/'> = 'PRSD' ; SS2<'('> = 'PRAL'
74 SS2<'('> = 'PRR' ; SS2<'.'> = 'PRCU'
76 SS2<'*'> = 'PREND' ; SS2<'>'> = 'PRGT'
78 SS2<'*'> = 'PRFD' ; SS2<'<'> = 'PRLT'
80 UT<'>'> = 1 ; OT<'LT'> = 1 ; OT<'LE'> = 1 ; OT<'NOT'> = 1
84 OT<'AND'> = 1 ; OT<'OR'> = 1 ; OT<'IMP'> = 1 ; OT<'EOV'> = 1
88 OT<'IN'> = 1 ; OT<'INSIDE'> = 1 ; OT<'EO'> = 1 ; OT<'GT'> = 1
92 UT<'IF'> = 1 ; UT<'ELSE'> = 1 ; UT<'UNION'> = 1
96 OT<'INST'> = 1
98 OT<'ST'> = 1 ; OT<'D'> = 1 ; OT<'THEN'> = 1 ; OT<'>'> = 1
100 OT<'NE'> = 1
101 XT<'>'> = 1 ; XT<'>-U'> = 1 ; XT<'>NOT'> = 1
104 XT<'>'> = 2 ; XT<'>-> = 2 ; XT<'>EO'> = 2 ; XT<'>NE'> = 2
108 XT<'>LT'> = 2 ; XT<'>LE'> = 2 ; XT<'>GT'> = 2 ; XT<'>GE'> = 2
112 XT<'>AND'> = 2 ; XT<'>OR'> = 2 ; XT<'>IMP'> = 2
115 XT<'>EOV'> = 2 ; XT<'>'> = 2 ; XT<'>'> = 2 ; XT<'>'> = 2
119 XT<'>UNION'> = 2 ; XT<'>INST'> = 2 ; XT<'>SD'> = 2
122 XT<'>INSIDE'> = 2
123 XT<'>ST'> = 2
124 XT<'>ARRAY'> = 3 ; XT<'>FA'> = 4 ; XT<'>SEI'> = 4
127 XT<'>IF'> = 4 ; XT<'>IF'> = 5 ; XT<'>THEN'> = 5
130 XT<'>IF'> = 4
131 XT<'>ELSE'> = 5
132 XT<'>SUM'> = 6 ; XT<'>PRDD'> = 6 ; XT<'>MAX'> = 6
135 XT<'>ALL'> = 6 ; XT<'>FALL'> = 6 ; XT<'>SET'> = 6
138 XT<'>'> = 15 ; XT<'>'> = 7 ; XT<'>'> = 8
141 XT<'>'> = 9 ; XT<'>'> = 10 ; XT<'>'> = 11
144 XT<'>'> = 12 ; XT<'>'> = 13 ; XT<'>'> = 14
147 IPK<'>-U'> = 23
149 IPK<'>-U'> = 23
149 IPK<'>'> = 24
150 IPK<'>'> = 27
151 IPK<'>'> = 27
152 IPK<'>'> = 26
153 IPK<'>'> = 26
154 IPK<'>'> = 25
155 IPK<'>'> = 25
156 IPK<'>'> = 25
157 IPK<'>'> = 25
158 IPK<'>'> = 25
159 IPK<'>'> = 24
160 IPK<'>'> = 24
161 IPK<'>'> = 24
162 IPK<'>'> = 21
163 IPK<'>'> = 20

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

164 IPR<'S1'> = 18
165 IPR<'$P'> = 18
166 IPR<'$S'> = 18
167 IPR<'$ST'> = 13
168 IPR<'SD'> = 16
169 IPR<'$INST'> = 15
170 IPR<'$UNION'> = 14
171 IPR<'EST'> = 13
172 IPR<'SEU'> = 12
173 IPR<'$NE'> = 12
174 IPR<'SGT'> = 12
175 IPR<'$GL'> = 12
176 IPR<'SLF'> = 12
177 IPR<'$LT'> = 12
178 IPR<'$INSIDE'> = 12
179 IPR<'$, ' > = 10
180 IPR<'$IN'> = 10
181 IPR<'$ALL'> = 7
182 IPR<'$FA'> = 7
183 IPR<'$EI'> = 7
184 IPR<'SEP'> = 7
185 IPR<'EF'> = 7
186 IPR<'ENT'> = 6
187 IPR<'AND'> = 5
188 IPR<'SUR'> = 4
189 IPR<'$IMP'> = 3
190 IPR<'$ENV'> = 2
191 IPR<'$( ' > = 1
192 PRETX = ARRAY('330')
193 MAXL = ARRAY('3')
194 OUTPUT('TITLE','SYSPRINT',(IHI,132A1))
195 PD = ' | ' . ' | ' . ' | RPOS(0) | '+' | '=' | ' | '*' | '/'
+   | '-' | '('
196 FD = '(' | ' | ' . ' | POS(0) | '-' | '+' | ' | '*' | '/'
+   | '='
197 DEFINE('AEVAL(EX)X,Y,Z,W,Y1,Z1,W1,PNT,IPW,IPW1,T')
198 DEFINE('CHECK(X,Y)')
199 DEFINE('SIMPAUL(EXPR)LTE,RTE')
200 DEFINE('PRINT(P1,P2)S1,S2,CUSR')
201 DEFINE('NDSpace(X)')
202 DEFINE('IN(Y)')
203 DEFINE('INI(Y)')
204 DEFINE('STOFORN(EXPR)S1,S2,S3,T1,T2,T3,S11,S21,S31,MAX,FOLD')
205 DEFINE('MACH')(X,MACTXT,MACODE,ARGLIST,MACNAM,ARGS,ARGNO')
206 DEF INE('MAXLENS(TEXT)I,ENTY')
* READ IN INFORMATION ABOUT THEOREMS
207 RDINF INFO = TRIM(INPJ)
208 IDENT(INFO,'1444444444') :S(READ)
* READ IN AND PRINT OUT THEOREMS
209 OUTPUT = INFO :(RDINF)
210 READ THEM = INI() :F(ENRD)
211 IDENT(THEM,'5555555555') :S(ENRD)
212 OUTPUT = THEM
213 THEM BREAK('') . TN :F(READ)
214 TN '**' =
215 THEM '**' =

```



```

216      S(TN) = THEM  : (READ)
217 ENRD
  * READ IN USER DEFINED MACROS
218      MACR()
  * READ IN ABBREVIATIONS USED IN ASSERTIONS
219 ST10A X = IN()
220      IDENT(X, '3333333333') :S(ST1)
221      OUTPUT = X
222      X BREAK('=') . X1 '=' REM . X2 =
223      S(X1) = X2 : (ST10A)
  * START THE PROVING OF VERIFICATION CONDITIONS
224 ST1  HYP0 = INI()
225      TST = TIME()
226      IDENT(HYP0, '6666666666') :S(ENR)
227      TITLE =
228      HYP0 SPAN('*') :F(A3)
229      OUTPUT = 'COMMENTS'
230 ST2  OUTPUT = HYP0
231      HYP0 = INI()
232      HYP0 SPAN('*') :S(ST2)
233 AC   OUTPUT =
234      OUTPUT = 'PROVING OF VERIFICATION CONDITIONS'
235      ASTNTBL = TABLE(10,10)
236      OUTPUT =
237      OUTPUT = 'HYPOTHESES'
238      HYP0 = NOSPACE(HYP0)
  * READ IN HYP
239      MAXL<1> = 0 ; MAXL<2> = 0 ; MAXL<3> = 0
242      IDENT(HYP0, '*') :S(A2)
243      IHC = 1
244      IDENT(HYP0) :S(A1)
245      HYP0 BAL . X ('*' | RPOS(0)) = :F(EXXOR14)
246      HYP = INFX(X(INTERFM(X)))
247      OUTPUT = 'H' IHC ': ' HYP
248      ASTNTBL<HYP> = 'H' IHC ': ' : (A1A)
249      HYP0 = INI()
250      IDENT(HYP0, '*') :S(A2)
251 A1A  IDENT(HYP0) :S(A1)
252      HYP0 BAL . X ('*' | RPOS(0)) = :F(EXXOR14)
253      Y = INFX(X(INTERFM(X)))
254      OUTPUT = DIFFER(ASTNTBL<Y>) ASTNTBL<Y> Y :S(A1A)
255      HYP = HYP '*' Y
256      IHC = IHC + 1
257      OUTPUT = 'H' IHC ': ' Y
258      ASTNTBL<Y> = 'H' IHC ': ' : (A1A)
  * READ IN CONC
259 A2   OUTPUT =
260      OUTPUT = 'CONCLUSIONS'
261      CONC =
262      IHC =
263      CONC = INI()
264      IDENT(CONC) :S(A3)
265      CONC BAL . X ('*' | RPOS(0)) = :F(EXXOR14)
266      Y = INFX(X(INTERFM(X)))
267      OUTPUT = DIFFER(ASTNTBL<Y>) ASTNTBL<Y> Y :S(A3A)
268      CONC = Y

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

269      IHC = IHC + 1
270      OUTPUT = 'C' IHC ': ' Y
271      ASTNTBL<CONC> = 'C' IHC ': '      : (A3A)
272 A3    CONO = IN()
273      IDENT(CONC, '*') :S(I1)
274 A3A    IDENT(CONC) :S(A3)
275      CONO BAL . X ('.' | RPOS(0)) = :F(EXXOR14)
276      Y = INFXX(INTERFM(X))
277      OUTPUT = DIFFER(ASTNTBL<Y>) ASTNTBL<Y> Y :S(A3A)
278      CONC = IDENT(CONC) Y :S(A4)
279      CONC = CONC 'Y'
280 A4      IHC = IHC + 1
281      OUTPUT = 'C' IHC ': ' Y
282      ASTNTBL<Y> = 'C' IHC ': '      : (A3A)

* READ IN THEOREMS USED IN THIS PROVING
283 I1      OUTPUT =
284      OUTPUT = 'THEOREMS USED'
285 I3      X1 = IN()
286      IDENT(X1, '*') :S(I4)
287 I3A    X1 DIFFER(X1) BAL . X ('.' | RPOS(0)) = :F(I3)
288      X 'X' =
289      LENE = X
290      X 'TAUT' BREAK(' ') '= RPOS(0) . TEXT :F(I2)
291      X = NOSPACE(X)
292      TEXT = INTERFM(TEXT)
293      TEXT = SFPP(TEXT)
294      WANG(' ' TEXT) :F(EXXOR11)S(I5)
295 I2      X = S(X)
296      IDENT(X) :S(EXXOR4)
297 I5      OUTPUT = X
298      X BREAK(' ') . TH1 RTAB(0) . TH2 :F(I4)
299      S('TH.' TH1) = TH2 : (I3A)

* PROVE THE VERIFICATION CONDITIONS
300 I4      OUTPUT =
301      OUTPUT = 'PROOF'
302      IL = 0
303      IA = 0
304      IV = 1
305      PRFTX<IV> = 'THEOREMS:HYPOTHESES:CONCLUSIONS'
306      MAXLEN(S(PRFTX<IV>))

* READ IN PROVING STEPS (THEOREM CALLS)
307 P1      LENE = NOSPACE(IN()) :F(FINISH)
308      IDENT(LENE, '8888888888') :S(FINISH)
309      LENE ('MACEXP(' | 'MACRO(' ID . X ') RPOS(0) :F(P1A)
310      IX = MACHC<X>
311      IDENT(IX) :S(EXXOR12)
312      EQEV = IDENT(MACTBL<IX,4>,'P') '.EQV.' :S(P1B)
313      EOFV = '.EQ.'
314 P1B      HL =
315      CL = INFXX(INTERFM(X EQEV MACTBL<IX,3>))) : (C4A)

* CHECK IF IMPLICATION EXPRESSION IS REQUIRED: THEOREM CALLS START WITH
* 'HH'.'H' OR 'HHH'
316 P1A      LENE 'HYPPXP(' BAL . X1 'EAL . X2 ') RPOS(0) :F(P1C)
317      DIFFER(ASTNTPL<X1>) :F(EXXOR13)
318      CHECK(X1,X2) :F(EXXOR13)

```

```

319 HL =
320 CL = INFX(X INTERFM(X2)) : (C9A)
321 PIC AXTAG = 0
322 LENE '###' = :F(P7)
323 AXTAG = 3 : (P5)
324 P7 LENE '##' = :F(P4)
325 AXTAG = 2 : (P5)
326 P4 LENE '#' = :F(P5)
327 AXTAG = 1
328 P5 LENE BREAK('(') . TH '(' RTAB(1) . APL ')' :F(EXXOR1)
329 TH = $('TH' TH)
330 IDENT(TH) :S(EXXOR6)
331 TH '(' ARG . FPL ')' = RTAB(0) . HACC (NULL $ HACC)
+ :F(EXXOR7)
332 R6 FPLC = FPL
333 SEPARTBL = TABLE(5,5)
334 CANCHOR = 0
335 R0 FPLC DIFFER(FPLC) BAL . FP ('' | RPOS(0)) = :F(R4)
336 FP BREAK('(') . FN '(' BAL . FPL1 ')' RPOS(0) :F(R2)
337 RA FPL FD . X FN '(' = X '%' FN '(' :S(RA)
338 R1 HACC FD . X FN '(' = X '%' FN '(' :S(R1)F(R0)
339 R2 HACC FD . X FP RD . Y = X '%' FP Y :S(R2)
340 R2A FPL FD . X FP RD . Y = X '%' FP Y :S(R2A)F(R0)
341 R4 CANCHOR = 1
342 FQ(AXTAG,0) :S(R3)
343 HACC HAL . X .IMP. REM . Y :S(R5)
344 HACC HAL . X .EQV. REM . Y :F(EXXOR2)
345 HACC = EQ(AXTAG,3) INTERFM(HACC) :S(P2)
346 Z = FQ(AXTAG,2) X :F(R5)
347 X = Y
348 Y = Z
349 R5 HACC = INTERFM(X .IMP. Y) : (P2)
350 R3 HACC DIFFER(HACC) BAL . X ('' | .IMP. | .EQV. | RPOS(0)) . Y
+ :F(P2)
351 Y (.IMP. | .EQV.) RPOS(0) :S(R7)
352 HACC = HACC INTERFM(X) Y : (R3)
353 R7 HACC = HACC INTERFM(X) Y INTERFM(HACC) : (P2)
354 P2 APL DIFFER(APL) BAL . AP ('' | RPOS(0)) = :S(P6)
355 IDENT(FPL) :S(C1)F(EXXOR5)
356 P6 FPL DIFFER(FPL) BAL . FP ('' | RPOS(0)) = :F(EXXOR5)
357 AP = INTERFM(AP)
358 FP = INTERFM(FP)
359 FP ARBN1(HAL . ') . X0 $ARRAY RPOS(0) :F(P3)
360 AP = '(' AP ')'
361 LXL = 0
362 X0 ARBN1(BAL . ') . X0 INTEGER . YD . RPOS(0) :F(EXXOR5)
363 X0 ARBN1(BAL . ') . FPL1 '%' ID . W0 . RPOS(0) :F(EXXOR5)
364 P1 HACC (LXL(LXL) ARBN1(BALBKCO)) . L '%' W0 . Y0 $ARRAY REM . R
+ :F(P2)
365 APL1 =
366 Z0 = Y0
367 F1A L0 = L
368 L =
369 F1B L0 BALBKCU . X = :F(EXXOR5)
370 L = DIFFER(L0) L X :S(F1B)
371 APL1 = X APL1

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

372      Z9 = NE(ZC.1) ZC - 1 :S(F1A)
373      TEMP = AP
374      TFPL1 = FPL1
375 F2    TFPL1 DIFFER(TFPL1) BAL . FP1 . * = :F(F4)
376      APL1 DIFFER(APL1) LAL . AP1 . * = :F(EXXOR5)
377      FP1 = SFPAPTHL<FP1>
378      GANCHOR = 0
379 F5    TEMP DELIM . X FP1 DELIM2 . Y = X *%* FP1 Y :S(F5)
380 F3    TEMP DLLIM . X *%* FP1 DELIM2 . Y = X AP1 Y :S(F3)
381      GANCHOR = 1 :S(F2)
382 F4    HAC = L TEMP R :S(F1)
383      LXL = SIZE(L TEMP) + 1
384 P3    GANCHOR = 0
385      SFPAPTHL<FP> = AP
386      AP = '(* AP *)'
387 P3C   HAC DELIM . X FP DELIM2 . Y = X AP Y :S(P30)
388      GANCHOR = 1 :S(P2)
389 C1    GANCHOR = 0
390 C1A   HAC ('(' | ')') = :S(C1A)
391      GANCHOR = 1
392      FQ(AXTAG,C) :S(C4)
393      HAC = INFXX(HAC)
394      IL = IDENT(ASNTBL<HAC>) IL + 1 :S(C3)
395      ASNTBL<HAC> = 'L' IL ' '
396 C3    IV = IV + 1
397      PRFTX<IV> = '#* LENE *$NONE$* ASNTBL<HAC> HAC *$*
398      STDFORM(PRFTX<IV>) :S(F1)
399 C4    HAC BAL . HL .IMP.* REM . CL :S(C6)
400      HAC BAL . HL .EQV.* REM . CL :S(C15)
401      HL =
402      CL = INFXX(HAC)
403 C4A   IL = IDENT(ASNTBL<CL>) IL + 1 :S(C5)
404      ASNTBL<CL> = 'L' IL ' '
405 C5    IV = IV + 1
406      ASNTBL<CL> = 'C' = '*C'
407      PRFTX<IV> = LENE *$NONE$* ASNTBL<CL> CL *$*
408      STDFORM(PRFTX<IV>) :S(C12)
409 C15   IHL = HL
410      IHL DIFFER(IHL) EAL . H ('* | RPOS(0)) = :S(C6)
411      H = INFXX(H)
412      ASNTBL<H> HD :S(C13)
413      SIMPAGL(H) :S(C13)
414      H BAL . FX .EQ.* REM . FY :S(C131)
415      H = FY .EQ.* FX
416      ASNTBL<H> HD :S(C13)
417 C131  HD = HL
418      HL = CL
419      CL = MPD
420 C6    HL DIFFER(HL) BAL . H ('* | RPOS(0)) =
421      H = INFXX(H)
422      ASNTBL<H> HD :S(C7)
423      SIMPAGL(H) :S(C9)
424      H BAL . FX .EQ.* REM . FY :S(EXXOR3)
425      H = FY .EQ.* FX
426      ASNTBL<H> HD :S(EXXOR3)
427 C7    CL = INFXX(CL)

```

ORIGINAL PAGE IS
A POOR QUALITY

```

428      IL = IDENT(ASNTBL<CL>) IL + 1      :F(C8)
429      ASNTBL<CL> = 'L' IL ':'
430      IV = IV + 1
431      ASNTBL<CL>      'C' = '*C'
432      PRFTX<IV> = LENE '3' ASNTBL<H> H '5' ASNTBL<CL> CL '5'
433      STDFORM(PRFTX<IV>)      : (C10)
434      IA = IA + 1
435      ASNTBL<H> = 'A' IA ':'      : (C7)
436      HL DIFFER(HL)      BAL . H ('.' | RPOS(0)) =      :F(C12)
437      H = INFXX(H)
438      ASNTBL<H> HD      :S(C11)
439      SIMPAGL(H)      :S(C101)
440      H BAL . FX '*EQ*' REM . FY      :F(EXXOR3)
441      H = FY '*EQ*' FX
442      ASNTBL<H> HD      :S(C11)F(EXXOR3)
443      IA = IA + 1
444      ASNTBL<H> = 'A' IA ':'
445      IV = IV + 1
446      IDENT(ASNTBL<H> ASNTBL<CL>)      :S(EXXOR3)
447      PRFTX<IV> = '5' ASNTBL<H> H '55'
448      STDFORM(PRFTX<IV>)      : (C10)
449      HYP = IDENT(HYP) CL      :S(P1)
450      HYP = HYP '*' CL      : (P1)

```

```

*      OUTPUT THE THEOREM CALLS, HYPOTHESES, AND CONCLUSIONS OF EACH
*      PROVING STEP IN THE STANDARD FORMAT
*

```

```

451 ENDPATHO      EO = 1
452 FINISH      OUTPUT =
453      I = 1
454 PRT0      J = 1
455      SINE =
456 PRT1      FNTY =
457      PRFTX<I> BREAK('5') . ENTY '5' =      :F(PRT5)
458      QX = DUPL(' ' MAXL<J> + 3)
459      QX LFN(SIZE(ENTY)) = ENTY
460      SINE = SINF QX
461      J = LT(J,3) J + 1      :S(PRT1)
462      OUTPUT = SINE
463 PRT3      I = LT(I,IV) I + 1      :S(PRT3)
464 PRT5      EO = EQ(EO,1) 0      :F(PRT4)
465      OUTPUT =
466      OUTPUT = EHRMSG
467      OUTPUT =
468      OUTPUT = '***** THEOREM NOT PROVED *****'      : (ENDPATH)
469 PRT4      TEMP = CONC
470 E1      TEMP      BAL . C '*' REM . H2      :F(E3)
471 E4      CC = HYP
472 E6      CC      BAL . G1 '*' REM . G2      :F(E9)
473      IDENT(G1,C)      :S(E7)
474      CC = G2      : (E6)
475 E7      TEMP = DIFFER(TEMP) H2      :F(OK)S(E1)
476 E9      IDENT(CC,C)      :S(E7)
477      SIMPAGL(C)      :F(E10)
478      OUTPUT = '      * ASNTBL<C> C * IS AN AXIOM'      : (E7)
479 E10      TT = 1

```

```

480      OUTPUT =
481      OUTPUT = 'UNPROVED CONCLUSION--' C  :{E7}
482 E3      C = TEMP
483      TEMP =  :{E4}
484 OK      OUTPUT =
485      IT = EU(IT,1) 0  :F(OK1)
486      OUTPUT = '***** THEOREM NOT PROVED *****'  :{ENDPATH}
487 OK1     OUTPUT = '***** Q. E. D. *****'
488      TIMES = TIME() - TST
489      TIMES RTAB(3) . X = X '.*'
490      :{ENDPATH}
      *      OUTPUT = '***** TIME TAKEN: ' TIMES ' SEC. *****'  :{ENDPATH}
      *
      * ERROR MESSAGES OR ERROR HANDLING IN THIS PROVING STEP
      *
491 ENDPATH IDENT(LENE,'8888888888')  :S(OUTP)
492          IDENT(LENE,'6666666666')  :S(UPDATE0)
493          LENE = IN()
494          OUTPUT = LENE  :{ENDPATH}
495 OUTP     OUTPUT =
496          OUTPUT =
497          OUTPUT =
498          OUTPUT =  :{ST1}
499      FOEV = IDENT(MACTBL<IX,4>,'P') 'EQV.'
500 EXXOR1   ERRMSG = 'FORMAT ERROR IN THE CALL STMT--' LENE  :{ENDPATH}
501 EXXOR2   ERRMSG = 'MISS --IMP-- SYMBCL IN THE EXPRESSION--'
      *      HAC  :{ENDPATH}
502 EXXOR3   ERRMSG = 'PREMISE NOT SATISFIED--' H  ' IN THEOREM '
      *      'CALL ' LENE  :{ENDPATH}
503 EXXOR4   OUTPUT = 'NO THEOREM BY THIS NAME OR ERROR IN INPUT SET-UP'
      *      '---' LENE  :{ENDPATH}
504 EXXOR5   ERRMSG = 'ACTUAL AND FORMAL PARAMETERS NOT IN CORRESPONDENCE'
      *      'APL--' APL 'FPL--' FPL 'IN THEOREM CALL ' LENE
      *      :{ENDPATH}
505 EXXOR6   ERRMSG = 'NO THEOREM BY THIS NAME IN THE THEOREM REPERTOIRE--'
      *      '---' LENE  :{ENDPATH}
506 EXXOR7   OUTPUT = 'FORMAT ERROR IN THEOREM--' TH
507          OUTPUT = '***** THEOREM NOT PROVED *****'  :{ENDPATH}
508 EXXOR8   OUTPUT = 'SYNTAX ERROR IN--' LINE ' IN SUBROUTINE--' SUBRTN
      *      :{FRETURN}
509 EXXOR10  OUTPUT = 'ERROR IN INFIX TRANSFORMATION--' PRINTLINE :{FRETURN}
510 EXXOR11  OUTPUT = 'SYNTAX ERROR IN TUATLOGY: ' TEXT  :{EXXOR}
511 EXXOR12  ERRMSG = 'UNDEFINED MACPC NAME IN THE CALL STMT--' LENE
      *      :{ENDPATH}
512 EXXOR13  OUTPUT = X1 ' IS NOT AN ABBREVIATION DEFINED OR '
      *      X2 ' IS NOT IN THE SPECIFIED ABBREVIATION'  :{ENDPATH}
513 EXXOR14  OUTPUT = 'SYNTAX ERROR IN HYPOTHESES OR CONCLUSIONS'
      *      :{ENDPATH}
      *      *****
      *      DEFINE('INI(Y)')
514 IN      Y = TRIM(INPUT)  :F(FRETURN)
515          Y LEN(79) . Z 'C' = Z  :F(IN)
516          IN = IN NOSPACE(Y)  :{IN}
517 IN      IN = IN NOSPACE(Y)  :{RETURN}
      *      *****
      *      DEFINE('INI(Y)')

```

```

*****
518 IN1 Y = TRIM(INPUT) :F(FRETURN)
519 Y LEN(79) . Z *C* = Z :F(INA)
520 IN1 = IN1 Y : (IN1)
521 INA IN1 = IN1 Y : (RETURN)
*****
*****
* DEFINE('MACR()X,MACTXT,MACODE,ARGLIST,MACNAM,ARGS,ARGNO')
*
* MACOR IS THE TABLE WHICH CONTAINS MACRO NAMES.
* MACTXL IS THE ARRAY WHICH CONTAINS # OF ARGUMENTS, MACRO ARGUMENT LIS T, AND
* MACRO DEFINITION
522 MACR MACX = 0
523 TITLE =
524 OUTPUT = '--- USER DEFINED MACROS ---'
525 OUTPUT =
526 MACRO = TABLE(100,30)
527 MACTXL = ARRAY('200,4')
528 TYPE = 'P'
529 MOC MACTXT = IN()
530 OUTPUT = MACTXT
531 IDENT(MACTXT,'2222222222') :S(RETURN)
532 TYPE = IDENT(MACTXT,'1111111111') *E* :S(MOO)
533 MACTXT BAL . X *E* REM . MACCODE :F(%E 'MO1')
534 MO4 MACX = MACX + 1
535 MACRO<X> = MACX
536 MACTXL<MACX,4> = TYPE
537 MACTXL<MACX,3> = MACCODE : (MOO)
*****
* DEFINE('PRINT(P1,P2)S0,S1,S2,CUSR')
*****
538 PRINT OUTPUT = P2
539 PRINTC CUSR = 0
540 PRINT1 P1 (TAB(CUSR) BREAK(' ')) . S1 ' ' REM . S2 :F(PRINT3)
541 S1 (BREAK(' ') BAL) . SC | (BAL . S0)
542 OUTPUT = IDENT(S1,S0) S1 :F(PRINT2)
543 P1 = S2 : (PRINT0)
544 PRINT2 CUSR = SIZE(S1) + 1 : (PRINT1)
545 PRINT3 OUTPUT = P1 : (RETURN)
*****
546 MAXLENS I = 1
547 MS1 TEXT DIFFER(TEXT) BREAK(' ') . ENTY 'S' = :F(RETURN)
548 MAXL<I> = GT(SIZE(ENTY),MAXL<I>) SIZE(ENTY)
549 I = I + 1 : (MS1)
*****
* DEFINE('STDFORM(EXPR)S1,S2,S3,T1,T2,T3,S11,S21,S31,MAX,FOLD')
*****
550 STDFORM EXPR BREAK(' ') . T1 ' ' BREAK(' ') . T2 'S' BREAK(' ') . T3
551 S1 = SIZE(T1) ; S2 = SIZE(T2) ; S3 = SIZE(T3)
552 MAX = S1 ; MAX = GT(S2,S1) S2 ; MAX = GT(S3,MAX) S3
553 FOLD = MAX / 41 + 1
554 EQ(FOLD,1) :S(SM4)
555 T1 LE(SIZE(T1),41) REM . S11 = :S(SM2)
556 T1 LEN(41) . S11 =
557 T2 LE(SIZE(T2),41) REM . S21 = :S(SM3)
558 T2 LEN(41) . S21 =

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

563 SM3      T3 LE(SIZE(T3),41) REM . S31 = :S(SM5)
564          T3 LEN(41) . S31 = '
565 SM5      PRFTX<IV> = S11 . S21 . S31 . S3
566 SM4      MAXLFS(PRFTX<IV>)
567          FOLD = FOLD - 1
568          IV = GT(=OLD,0) IV + 1 :S(SM1)F(RETURN)
*****
*          DEFINE('NOSPACE(X)')
*****
569 NOSPACE  CANCHOR = 0
570 NOSPACE  X SPAN(' ') = :S(NOSPACEC)
571          NOSPACE = X
572          CANCHOR = 1 : (RETURN)
*****
*          DEFINE('CHECK(X,Y)')
*          THIS ROUTINE CHECKS IF Y IS IN X(X)
*****
573 CHECK    SAVLD = CANCHOR
574          CANCHOR = 0
575          $(X) (POS(G) | ' ') Y (' ' | RPOS(0)) :F(CHECK1)
576          CANCHOR = SAVLD : (RETURN)
577 CHECK1    CANCHOR = SAVLD : (RETURN)
*****
*          DEFINE('SIMPAGL(EXPR)LTE,RTE')
*****
578 SIMPAGL   EXPR BAL . LTE ' ' RELTN ' ' BAL . RTE RPOS(0) :F(FRETURN)
579          IDENT(RENA,'EU') IDENT(LTE,RTE) :S(RETURN)
580          LTE POS(0) SPAN('0123456789()*/+-BO') RPOS(C) :F(FRETURN)
581          RTE POS(0) SPAN('0123456789()*/+-BO') RPOS(0) :F(FRETURN)
582          LTE = INTERFM(LTE)
583          RTE = INTERFM(RTE)
584          LTE = AEVAL(LTE) :F(FRETURN)
585          RTE = AEVAL(RTE) :F(FRETURN)
586          $(KENA) '(* LTE ' ' RTE ' ') :S(RETURN)F(FRETURN)
587          *****
*          DEFINE('AEVAL(EX)X,Y,Z,W,Y1,Z1,W1,PNT,IPW,IPW1,T')
*****
588 AEVAL     OPNSTK = ARRAY('20')
589          PNT = 0
590 AL0       IDENT(EX) :S(AL9)
591          EX (BREAK(' ') . X ' ') | REM . X =
592          X POS(0) 'S' = :S(AL1)
593          PNT = LT(PNT,20) PNT + 1 :F(AL11)
594          X POS(0) '0' SPAN(DIGIT) . Y RPOS(0) :F(AL0A)
595          OCTAL = 0
596 AL0B      Y DIFFER(Y) LEN(1) . Z = :F(AL0C)
597          OCTAL = OCTAL * 8 + Z : (AL0B)
598 ALCC      X = OCTAL 'B23'
599 ALCA      OPNSTK<PNT> = X : (AL0)
600 AL1       IDENT(X,'+U') :S(AL0)
601          X = IDENT(X,'-U') ' - ' :F(AL5)
602          OPNSTK<PNT> BREAK('B') . Y 'B' REM . Z = :F(AL2)
603          OPNSTK<PNT> = '(-' Y 'B') Z : (AL0)
604 AL2       OPNSTK<PNT> = -OPNSTK<PNT> : (AL0)
605 AL5       PNT = PNT - 1

```

ORIGINAL PAGE IS
OF POOR QUALITY

.00000043
 .00000046
 .00000044
 .00000031
 .00000032
 .00000033
 .0
 .00000036
 .00000037
 .00000038
 .00000039
 .00000040
 .00000041
 .00000045
 .00000042
 .000000310
 .000000320
 .000000330
 .000000340
 .000000350
 .000000370
 .
 .
 .
 .
 .
 .000000390
 .000000400
 .000000410
 .000000420
 .000000430
 .000000440

```

606      X = ' * X * '      :S(ALB)      .00000450
607      IDENT(X, ' * * ') :S(ALB)      .00000460
608      OPNSTK<PNT> BREAK('B') . Y 'B' REM . Z = :S(AL6) .00000470
609      OPNSTK<PNT + 1> BREAK('B') . Y 'B' REM . Z = :S(ALB) .00000480
610      T = OPNSTK<PNT>      .00000490
611      IPW = 23 - Z      .00000500
612      IDENT(X, ' / ') :S(AL5A)      .00000510
613      W = EVAL(Y ' * 2 * * ' IPW) :F(FRETURN) .00000520
614      OPNSTK<PNT> = EVAL(T X W) 'B23' :S(AL0)F(FRETURN) .00000521
615      AL5A W = EVAL(T ' * 2 * * ' IPW) :F(FRETURN) .00000530
616      OPNSTK<PNT> = EVAL(W X Y) 'B23' :S(AL0)F(FRETURN) .00000540
617      AL6 OPNSTK<PNT + 1> BREAK('B') . Y1 'B' REM . Z1 = :S(AL6A) .00000550
618      T = OPNSTK<PNT + 1>      .00000560
619      IPW = 23 - Z      .00000570
620      W = EVAL(Y ' * 2 * * ' IPW) :F(FRETURN) .00000580
621      OPNSTK<PNT> = EVAL(Y X T) 'B23' :S(AL0)F(FRETURN) .00000590
622      AL6A IDENT(X, ' * ') :S(AL6C)      .00000600
623      IPW = 23 - Z      .00000610
624      IPW1 = 23 - Z1      .00000620
625      W = EVAL(Y ' * 2 * * ' IPW)      .00000630
626      W1 = EVAL(Y1 ' * 2 * * ' IPW1)      .00000640
627      IDENT(X, ' / ') :S(AL6B)      .00000650
628      OPNSTK<PNT> = EVAL(W X W1) 'B23' :S(AL0)F(FRETURN) .00000660
629      AL6B OPNSTK<PNT> = EVAL(W X W1) 'B0' :S(AL0)F(FRETURN) .
630      AL6C IPW = 23 - Z1 - Z      .
631      W = EVAL(Y X Y1)      .00000690
632      OPNSTK<PNT> = EVAL(W ' * 2 * * ' IPW) 'B23' :S(AL0)F(FRETURN) .00000700
633      ALB OPNSTK<PNT> = EVAL(OPNSTK<PNT> X OPNSTK<PNT + 1>) :S(AL0) .00000710
        + F(FRETURN)      .00000712
634      AL5 X = EQ(PNT,1) OPNSTK<1> :F(FRETURN) .00000720
635      X BREAK('B') . X1 'B' REM . Y1 :F(AL10) .00000730
636      X = EVAL(X1 ' * 2 * * (23 - ' Y1 ')) :S(AL10)F(FRETURN) .00000740
637      AL10 AEVAL = CONVERT(X, 'INTEGER') :S(RETURN)F(FRETURN) .00000750
638      AL11 OUTPUT = 'ERROR IN AEVAL ROUTINE; STACK OPNSTK IS OVERFLOWED' .00000760
        + :F(FRETURN)      .00000770
        *****
        * DEFINE('NOPARA(TXT)X1')
        *****
639      NOPARA TXT POS(C) ' ( ' JAL . TXT ' ) ' RPCS(C)
640      NOPARA = TXT :F(FRETURN)
        *****
        * DEFINE('INTERFM(X)Z,Y,L,H,OP,XO,HH,XH,COMMAS')
        *****
641      INTERFM
642      XO = X
643      X = X 'H'
644      L = C
645      H = 0
        ***
        ***
646      SCN1 X LEN(1) . Z =
647      Y = SS1<Z>
648      IDENT(Y) :S(ESSOR1)F(Y)
649      SCN1A STR<L> = Z Y :S(SCN2)
650      PRL L = L + 1
651      Y =

```

ORIGINAL PAGE

```

652      X SPAN(AN) * Y =
653      X * (' = ' :F(SCN1A)
654      H = H + 1
655      L = L - 1
656      STK<H> = '% Z Y' : (PRAL)
657 PRD  L = L + 1
658      Y =
659      X (ITRO (('.' ITRO 'B' ITR) | NULL)) * Y = : (SCN1A)
660 PRUP  OP = '+U' : (PROP)
661 PRUM  OP = '-U' : (PROP)
662 PRLP  H = H + 1
663      STK<H> = '%(' : (SCN1)
664 PRNULL IDENT(STK<H>,'%(' : F(ESSOR1)
665      H = H - 1
666      STK<L> = '%S1' : (SCN2)
667      L = L + 1
668 PRN  X 'IF' : S(PRP)
669      X (ITR 'B' ITR) * Y = : S(PRN2)
670      X ('TRUE' | 'FALSE') * Y = : F(PRN0)
671      L = L + 1
672      STK<L> = 'Y' : (SCN2)
673 PRN0  X 'NOT' = : F(ESSOR1)
674 PRN1  OP = 'NOT' : (PROP)
675 PRN2  L = L + 1 : (SCN1A)
676 PRNOT  X '=' = : F(PRN1)
677      OP = 'NE' : (PRQP)
678 PRNOT1 X '=' = : F(ESSOR1)
679      OP = 'NE' : (PROP)
680
681 ***
682 ***
680 SCN2 X LEN(1) * Z =
681      Y = S52<Z>
682      IDENT(Y) : S(ESSOR1)F($Y)
683 PRP  X BREAK('.' ) * OP '.' = : F(ESSOR1)
684      IDENT(OT<OP>) : S(ESSOR1)
685      OP = IDENT(OP) '...'
686      OP = IDENT(STK<H>,'%.' ) '....'
687 PRCP  OP = '% OP
688 PRCP0  EQ(H) : S(PROP1)
689      IDENT(OP,'$NOT') : S(PROP1)
690      STK<H> IDENT(OP,'$IF') ('$THEN' | '$ELSE') RPOS(0) : S(PROP1)
691      IDENT(STK<H>,'%IF') LT(IPR<OP>,24) : S(PROP1)
692      IDENT(OP,'$THEN') : F(PROPA)
693      IDENT(STK<H>,'%IF') : S(PROP1)F(PROP2)
694 PROPA  IDENT(OP,'$ELSE') : F(PROPB)
695      IDENT(STK<H>,'%THEN') : S(PROP1)F(PROP2)
696 PROPB  IDENT(OP,'$IN') IDENT(STK<H>,'%') : S(PROP1)
697      IDENT(OP,'%') IDENT(STK<H>,'%IN') : S(PROP1)
698      OP IDENT(STK<H>,'%') LUP RPOS(0) : S(PROP1)
699      STK<H> IDENT(OP,'%') SYFTN RPOS(0) : S(PROP1)
700      IDENT(OP,'%') IDENT(STK<H>,'%') : S(PROP1)
701      GT(IPR<OP>,'$PR<STK<H>') : S(PROP1)
702      STK<H> ('$E1' | '$E1' | '$SUM' | '$PROD') RPOS(0) : F(PROP2)
703      IDENT(OP,'%') : S(PROP1)
704 PRCP2  L = L + 1
705      STK<L> = STK<H>

```

```

706      H = H - 1      : (PROP0)
707 PROP1 H = H + 1
708      STK<H> = OP
709      OP = 'S...' FPOS(0)      :S(PRSET)F(SCN1)
710 PRSET STK<H - 1> = IDENT(STK<H - 1>,'S(') *$SET(' :S(SCN1)F(ESSOR1)
711 PRPO OP = Z      : (PROP)
712 PRT  X 'S' =      :F(PRBO)
713      OP = 'S...'      : (PRPO)
714 PRAL STK<H> 'S' ('FA' | 'EI' | 'EP' | 'SET' | FUNC | 'F1') RPOS(0)
+      :S(SCN1)
715      STK<H> 'S' =
716      H = H + 1
717      STK<H> = '$ARRAY('      : (SCN1)
718 PRGT X 'S' =      :S(PRGE)
719      OP = 'GT'      : (PRPO)
720 PRGE OP = 'GE'      : (PRPO)
721 PRLT X 'S' =      :S(PHLE)
722      OP = 'LT'      : (PRPO)
723 PRLE OP = 'LE'      : (PRPO)
724 PRGE OP = 'EO'      : (PRPO)
725 PRCO OP = 'S'      : (PRPO)
726 PRR  SH = STK<H>
727      SH 'S' FUNC * OP RPOS(0)      :F(PRR0)
728      STR<L + 1> = OP
729      STR<L + 2> = 1
730      STR<L + 3> = '$ARRAY'
731      L = L + 3
732      H = H - 1      : (SCN2)
733 PRR0
734      SH ('S' | '$IN' | 'S...' | 'S...' | 'S(' | '$ARRAY(') RPOS(0)
+      :S('$('PRR' XT<SH>))
735      L = L + 1
736      STR<L> = STK<H>
737      H = NE(H,1) H - 1      :S(PRR)F(ESSOR1)
738 PRR9 COMMAS = 0
739 PRR91 COMMAS = IDENT(STK<H>,'S(') COMMAS + 1      :F(PRR92)
740      H = H - 1      : (PRR91)
741 PRR92 H = IDENT(STK<H>,'S(') H - 1      :S(PRR95)
742      L = IDENT(STK<H>,'$ARRAY(') L + 1      :F(PRR93)
743      STR<L> = STK<H - 1>
744      H = NE(H,1) H - 2      :F(ESSOR1)
745      L = L + 1
746      STR<L> = COMMAS + 1
747      L = L + 1
748      STR<L> = '$ARRAY'      : (SCN2)
749 PRR93 H = IDENT(STK<H>,'$IN') H - 1
750      H = IDENT(STK<H>,'S(') H - 1      :F(PRR94)
751      COMMAS = COMMAS + 1      : (PRR93)
752 PRR94 STK<H> 'S' ('FA' | 'EI' | 'EP' | 'F1' | 'SUM' | 'PROD') RPOS(0)
+      :F(ESSOR1)
753      L = L + 1
754      STR<L> = COMMAS
755      L = L + 1
756      STR<L> = STK<H>
757      H = H - 1      : (SCN2)
758 PRR95 HH = L

```

```

759     L = L + COMMAS
760     LL = L
761     PRR951 STR<LL> = '$AND'
762     STR<LL - 1> = STR<HH>
763     COMMAS = NE(COMMAS,1) COMMAS - 1 :F(SCN2)
764     HH = HH - 1
765     LL = LL - 2 : (PRR951)
766     PRR10 COMMAS = 0
767     H = H - 1
768     PRR101 H = IDENT(STK<H>,'$') H - 1 :F(ESSOR1)
769     COMMAS = COMMAS + 1
770     H = IDENT(STK<H>,'$IN') H - 1 :S(PRR101)
771     STK<H> '$' ((' | 'SET' | FUNC) RPOS(0) :F(ESSOR1)
772     PRR102 L = L + 1
773     STR<L> = COMMAS
774     L = L + 1
775     STR<L> = IDENT(STK<H>,'$') '$ALL' :S(PRR103)
776     STR<L> = STK<H>
777     PRR103 H = H - 1 : (SCN2)
778     PRR11 COMMAS = 0 : (PRR92)
779     PRR12 H = H - 1
780     STK<H - 1> '$' FUNC . DP RPOS(0) :F(SCN2)
781     L = L + 1
782     L = L + 1
783     STR<L> = DP
784     STR<L> = 1
785     L = L + 1
786     STR<L> = '$ARRAY'
787     H = H - 1 : (SCN2)
788     PRR13 H = H - 2
789     L = L + 1
790     STR<L> = IDENT(STK<H + 1>,'$SET') '$S2' :F(ESSOR1)S(SCN2)
791     PRR14 H = H - 3
792     L = L + 1
793     STR<L> = IDENT(STK<H + 1>,'$SET') IDENT(STK<H + 2>,'$') '$S3'
794     + :S(SCN2)F(ESSOR1)
794     PREND SH = STK<H>
795     EV(H) :S(DONE0)
796     L = L + 1
797     STR<L> = SH
798     H = NE(H,1) H - 1 :S(PREND)F(DONE0)
799     DONE0
800     INTERFM = GE(L,1) STR<L> :F(ESSOR1)
801     L = NE(L,1) L - 1 :F(RETURN)
802     PRXX INTERFM = STR<L> * * INTERFM
803     L = NE(L,1) L - 1 :S(PRXX)F(RETURN)
804     ESSOR1 OUTPUT = 'SYNTAX ERROR IN STRING ' X0 ' ; ERROR OCCURS BEFORE '
+ 'STHING ' X ' STK<H> = ' STK<H> ' OP = ' OP ' STR<L> = ' STR<L>
+ : (EXXOR0)
*****
* DEF INE(' INFXX(ST)H,E,SH,IPRE,SHM,AYN,SUB,QUEXP,N,INNO,ALL,L')
*****
P05 INFXX
806     H = 0
807     ST = ST * *
808     DL ST BREAK(' ') * * = :F(DONE1)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

809      E 'S'      :F(DL1)
810      SH = 'D' DIFFER(XT<E>) XT<E>      :S($SH)F(ESSCR2)
811 DL1      H = H + 1
812      COP<H> = 0
813      STK<H> = E      : (DL)
814 D1      IPRE = E
815      SH = STK<H>
816      SH VAR RPOS(C)      :S(D1A)
817      SH ANY(NUM) SPAN(NUM) RPOS(0)      :S(D1A)
818      SH = GE(IPR<E>,COP<H>) *(' SH ')
819 D1A      F 'S' UACP . E 'U' RPOS(?)      :F(D1B)
820      E = IDENT(E,'*')      : (D1C)
821 D1B      E = IDENT(E,'*NOT') *NOT.*
822 D1C      STK<H> = E SH
823      COP<H> = IPR<IPRE>      : (DL)
824 D2      IPRE = IPR<E>
825      SH = STK<H>
826      EQ(IPRE,22)      :S(D2B)
827      SH (VAR | INTEGER) RPOS(0)      :S(D2B)
828      DIFFER(SH,'TRUE.') DIFFER(SH,'FALSE.')      :F(D2B)
829      SH *(' HAL ') RPOS(0)      :S(D2B)
830      E ('S+' | 'S*') RPOS(0)      :F(D2A)
831      STK<H> (UACP | MUOP) RPOS(0) EQ(IPRE,COP<H>)      :S(D2B)
832 D2A      SH = GE(IPRE,COP<H>) *(' SH ')
833 D2B      H = H - 1
834      SHM = STK<H>
835      EQ(IPRE,22)      :S(D2C)
836      SHM (VAR | INTEGER | *.TRUE.* | *.FALSE.*) RPOS(0)      :S(D2C)
837      SHM *(' HAL ') RPOS(0)      :S(D2C)
838      SHM = GT(IPRE,COP<H>) *(' SHM ')
839 D2C      E LEN(1) =
840      E = IDENT(OPT<E>.) *E.*
841 D22      STK<H> = SHM E SH
842      STK<H> = E
843      COP<H> = IPRE      : (DL)
844 D3      N = STK<H>
845      AYN = STK<H> - 1>
846      SUB = STK<H> - 2>
847      H = H - 2
848 D3A      N = NE(N,1) N - 1      :F(D3B)
849      H = H - 1
850      SUB = STK<H> *(' SUB      : (D3A)
851 D3B      STK<H> = AYN *(' SUB ')
852      COP<H> = IPR<E>      : (DL)
853 D4      F 'S' RFM . F
854      QUEXP =
855      INNO = STK<H>
856      H = H - 2
857      QUEXP = STK<H> *(' STK<H> + 1> ')
858 D4A      INNO = NE(INNO,1) INNO - 1      :F(D4B)
859      H = H - 1
860      QUEXP = STK<H> *(' QUEXP      : (D4A)
861 D4B      STK<H> = F *(' QUEXP
862      COP<H> = IPR<E>      : (DL)
863 D5      DIFFER(E,'ELSE') DIFFER(E,'THEN')      :F(DL)
864      H = H - 2

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

865 STKCH> = '.IF.' STKCH> '.THEN.' STKCH + 1> '.ELSE.' STKCH + 2>
866 COPCH> = IPRCE> : (DL)
867 D6 : (D4)
868 D7 H = H - 1
869 STKCH> = '(. STKCH> '...' STKCH + 1> '.)'
870 COPCH> = IPRCE> : (DL)
871 D8 H = H - 2
872 STKCH> = '(. STKCH> '...' STKCH + 1> '...' STKCH + 2> '.)'
873 COPCH> = IPRCE> : (DL)
874 D9 E = 'SAND' : (D2)
875 D15 H = H + 1
876 COPCH> = IPRCE>
877 STKCH> = '()' : (DL)
878 DONE1 INFXY = STKCH> : (RETURN)
879 ESSOR2 OUTPUT = 'ERROR IN INFIX TRANSFORMATION--' ST
+ '!--' E ' IS THE UNDEFINED OPERATOR' : (EXXDR0)
*****
* DEFINE ('SFPF (TEXT) X1, X2, X3')
*****
880 SFPF TEXT BREAK ('S') . X1 'S' (UNCP | BINDP) . X2 = :F (EXXDR11)
881 X1 FPOS(1) . X1
882 X1 = IDENT(X2, 'NOT') X2 '(. X1 '.)' :S(SF1)
883 X1 BREAK (' ') . X1 ' ' REM . X3
884 X1 = X2 '(. X1 ' ' X3 '.)'
885 SF1 TEXT = DIFFER (TEXT) X1 TEXT :S(SFPF)
886 SFPF = X1 : (RETURN)
*****
* DEFINE ('WANG (ANTE, CONSEQ) PHI, PSI')
*****
887 WANG CANCHOR = 0
888 ANTE FORMULA = :S('A' OP)
889 CONSEQ FORMULA = :S('C' OP)
890 ANTE = ANTE ' '
891 CONSEQ = ' ' CONSEQ ' '
892 TEST ANTE ATOM = :F (OUTW)
893 CONSEQ A :F (TEST)
894 CANCHOR = 1 : (RETURN)
895 OUTW CANCHOR = 1 : (RETURN)
896 AND WANG (ANTE, CONSEQ ' ' PHI) :S (RETURN) F (RETURN)
897 AAND WANG (ANTE ' ' PHI ' ' PSI, CONSEQ) :S (RETURN) F (RETURN)
898 AOR WANG (ANTE ' ' PHI, CONSEQ) :F (RETURN)
899 WANG (ANTE ' ' PSI, CONSEQ) :S (RETURN) F (RETURN)
900 AIMP WANG (ANTE ' ' PSI, CONSEQ) :F (RETURN)
901 WANG (ANTE, CONSEQ ' ' PHI) :S (RETURN) F (RETURN)
902 AFDV WANG (ANTE ' ' PHI ' ' PSI, CONSEQ) :F (RETURN)
903 WANG (ANTE, CONSEQ ' ' PHI ' ' PSI) :S (RETURN) F (RETURN)
904 CNOT WANG (ANTE ' ' PHI, CONSEQ) :F (RETURN) F (RETURN)
905 CAND WANG (ANTE, CONSEQ ' ' PHI) :F (RETURN)
906 WANG (ANTE, CONSEQ ' ' PSI) :S (RETURN) F (RETURN)
907 COR WANG (ANTE, CONSEQ ' ' PHI ' ' PSI) :S (RETURN) F (RETURN)
908 CIMP WANG (ANTE ' ' PHI, CONSEQ ' ' PSI) :S (RETURN) F (RETURN)
909 CEUV WANG (ANTE ' ' PHI, CONSEQ) ' ' PSI :F (RETURN)
910 WANG (ANTE ' ' PSI, CONSEQ) ' ' PHI :S (RETURN) F (RETURN)
911 END

```

ORIGINAL PAGE IS
OF POOR QUALITY